

AÑO 1 - NUMERO 3.

REVISTA EXCLUSIVA PARA USUARIOS

Código màquina para ampliar el BASIC

VU-3D: Trabajar en tres dimensiones

Base de datos en microdrive

Programas: El frogger en BASIC

ZX Spectrum +
(64 K.)
Para los que exigen +



Noviembre es un buen mes para todos los aficionados a la informática. Se celebra en Madrid el SIMO, donde tienen cabida todo tipo de ordenadores y donde estaremos como espectadores y como expositores a fin de informarles en torno al mundo del Spectrum, al igual que hicimos el mes pasado en Barcelona con motivo del SONIMAG y del que les damos buena cuenta en este número. No obstante la mayor novedad llegaría antes de finalizar esta feria, con la presentación del Spectrum plus, una nueva versión del Spectrum con teclado profesional, del que les daremos más información en el próximo número.

Pero noviembre es también un mes duro para cuantos trabajan sin descanso para poder tener listos sus programas para el SIMO en unos casos y para navidades en otros. El resultado a buen seguro que será muy positivo para todos los usuarios del Spectrum que podrán así disponer de una mayor oferta.

Número 3



El dibujo de portada ha sido realizado por el equipo de diseño de Dinamic Soft: Luis Rodríguez y Víctor Ruiz, con el programa Artist.

DIRECTOR:
Simeón Cruz
COORDINADOR EDITORIAL
J. A. Sanz
REDACCION:
Juan Arencibia
Fernando García
José C. Tomás
Gumersindo García
DISEÑO:
Ricardo Segura

Editado por
PUBLINFORMATICA, S.A.
PRESIDENTE:
Fernando Bolín
DIRECTOR EDITORIAL:
Norberto Gallego

Administración INFODIS, S. A.
CONSEJERO DELEGADO:
Fernando Bolín
GERENTE CIRCULACION Y VENTAS
Luis Carrero
PRODUCCION
Miguel Onieva
SERVICIO CLIENTES
Antonio Zurdo
JEFE DE PUBLICIDAD
María José Martín

Dirección:
Redacción y administración
C/ Bravo Murillo, 377.
Tel. 733 74 13
28020 Madrid

Publicidad Madrid:
Nieves Clemente
C/ Bravo Murillo, 377
Tel. 733 96 62
28020 Madrid

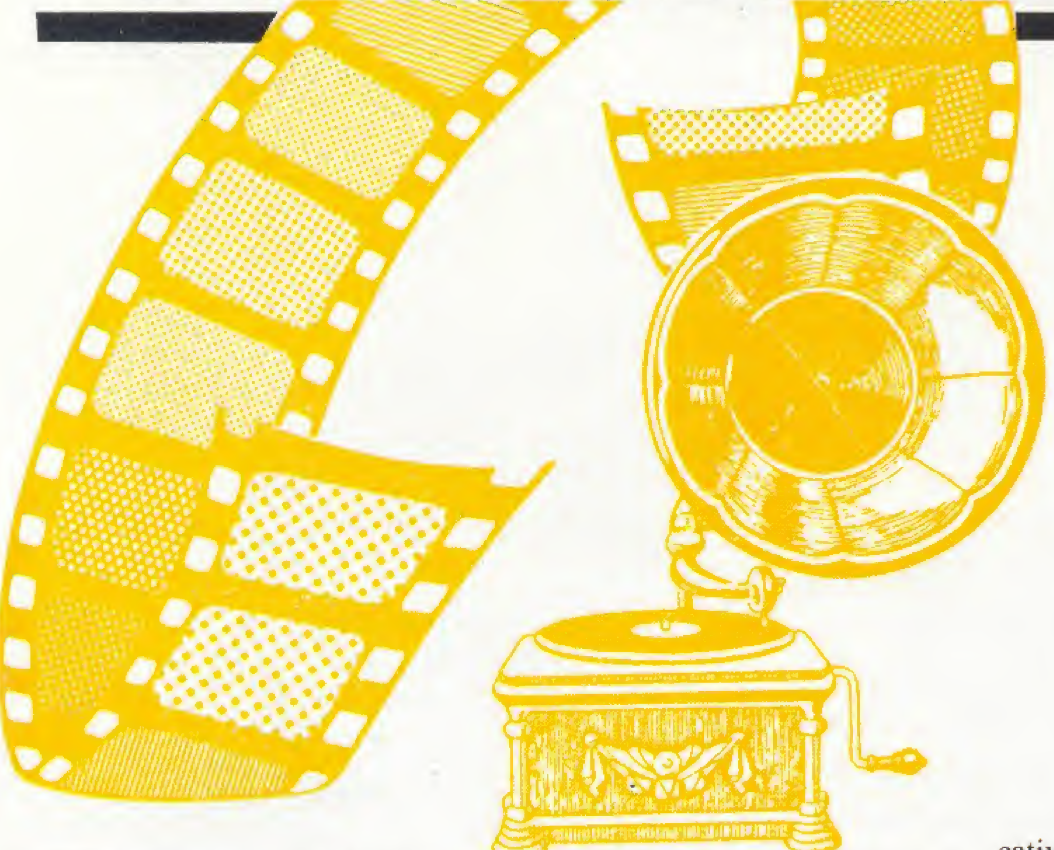
Publicidad Barcelona:
Roberto Rodríguez
C/Pelayo, 12
Tel. (93) 301 47 00 ext. 27
08001 Barcelona

Depósito Legal: M-29041-1984
Distribuye: S.G.E.L.
Avda. Valdelaparra, s/n.
Alcobendas. Madrid.

Fotomecánica: Karmat
C/ Pantoja, 10. Madrid.
Imprime: Héroes
C/ Torrelara, 8.
28036 Madrid

SUSCRIPCIONES
Rogamos dirija toda la correspondencia relacionada con suscripciones o números atrasados a:
TODOSPECTRUM
EDISA
LOPEZ DE HOYOS, 141, 5.º
28002-Madrid.

NOVEDADES



Un año más del 1 al 7 de octubre se celebró en Barcelona el salón de la imagen, sonido y electrónica, conocido por el Sonimag. Junto a tres grandes pabellones dedicados a los clásicos audio-visuales, con un llamativo retroceso en cuanto a la presencia de los radioaficionados, se encontraba un pequeño y acogedor pabellón dedicado al mundo de la informática, especialmente en cuanto a los ordenadores domésticos se refiere. Un buen lugar para grandes y chicos, pero especialmente para estos últimos, para quienes los distintos expositores posibilitaron el libro uso de los ordenadores.

ES en el campo del *software* donde las novedades se hicieron más patentes, pero tampoco faltaron periféricos de interés e incluso el nuevo QL de la casa **Sinclair,** que por primera vez caía en las manos del público español. Pero vayamos por partes a analizar las distintas novedades.

Decíamos que el *software* fue lo más llamativo. Pero no seríamos justos si no destacásemos la gran labor en materia de *software* educativo que vienen desarrollando activas casas de *software* catalanas, la mayoría de ellas de reciente aparición en el mercado. Una de estas empresas es **Idealogic,** quienes han adaptado el *software* de las multinacionales americanas **Spinnaker** y **Fisher Price** a los ordenadores



Spinnaker: programas educativos y divertidos.



El QL ya estaba al alcance de todos, aunque no todos parecían entenderle...

SONIMAG '84


Un sofisticado *interface* puede permitir al Spectrum la toma de datos externos



Dos libros más, preludio de la avalancha prevista para las próximas navidades.

Apple, Commodore y Spectrum. Los distintos programas presentados van dirigidos al público infantil con edades entre los 8 y los 14 años.

Con carácter igualmente educativo estaba el programa BASIC para las primeras lecciones de este lenguaje. Y el programa *Calc*, uno de los más completos que hemos visto en el campo de las matemáticas. *Profesor Dietético* era otra de las novedades especialmente recomendable para los que deseen vigilar más de cerca su alimentación. Estos tres programas eran presentados por **Software Centre**, empresa que aglutina a un nutrido número de jóvenes programadores. Esta misma empresa presentaba también dos paquetes de programas para el microdrive: *Fichero-agenda*, típico archivo de datos, y *Tecno-art* para la realización de dibujos con las facilidades de almacenamiento rápido en microdrive.

Tampoco faltaban los famosos *arcade games* o juegos de aventuras, en su mayoría traídos de las listas de éxito del mercado inglés. Este no era el caso del *Mad cars* realizado por **Ventamatic**, carrera de coches a través de un inmenso recorrido sobre campo a través, desiertos, puentes sobre el mar e incluso carreteras fantasmas. También la casa **Software Centre** apro-


Verbot: un pequeño robot que obedece a sus instrucciones verbales.

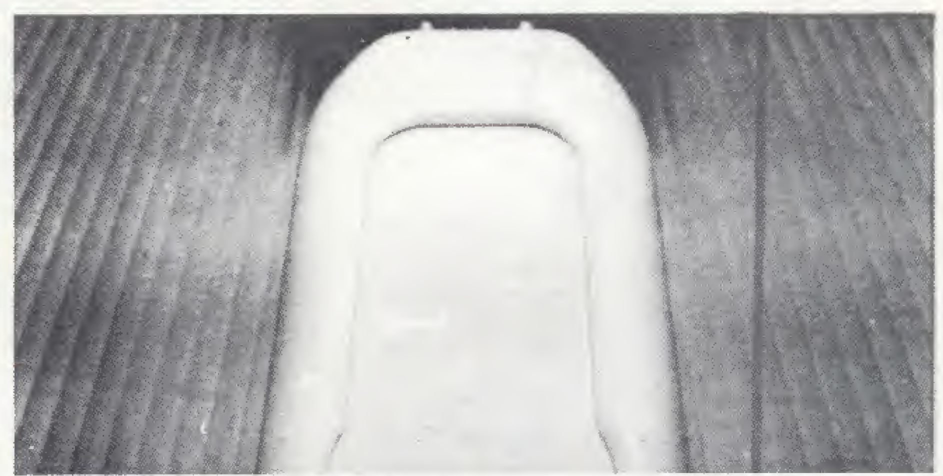


Profesor dietético. Una forma distinta de controlar su dieta.



vechaba la ocasión para presentar el programa *Juegos olímpicos*, que tenían la particularidad de utilizar un sintetizador de voz. Desgraciadamente habremos de esperar algunas semanas para su comercialización.

En el campo de las aplicaciones profesionales **Ventamatic** presentó las mayores novedades: *Context* V.6 es la versión sexta del *Context* basado en la originaria idea del *Tassword* que comentábamos en el número anterior. Otra de las aplicaciones «serias» fue el SITI o Sistema Integrado de Tratamiento de la Información. Base de datos de 10 Kbytes de memoria en código máquina que permite la realización de cálculos complejos entre los distintos campos de una ficha. Igualmente realizado por los programadores de esta conocida firma.

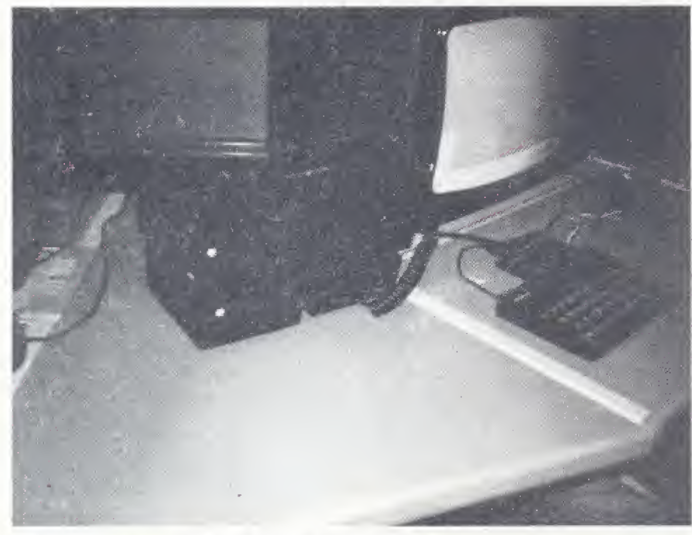


No busque los cables... este *joystick* funciona por rayos infrarrojos.

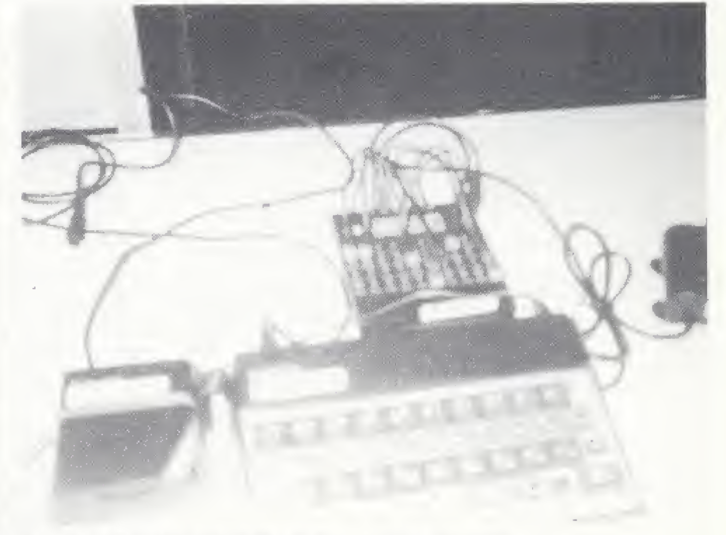


Cassette especial para ordenadores de fácil manejo.

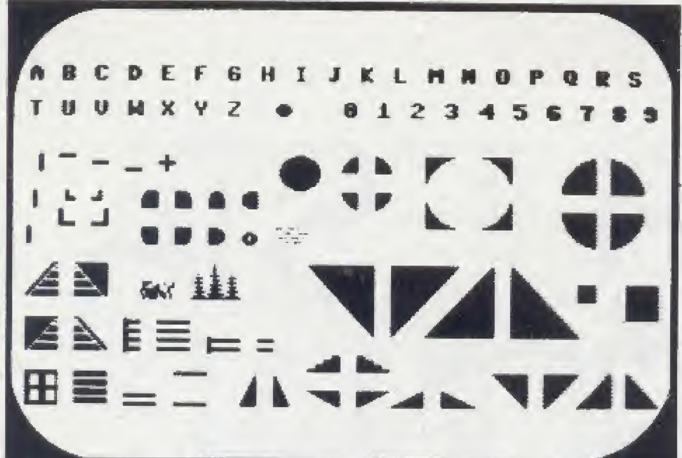
Las novedades *hardware* más importantes vendrían de la mano de **Investrónica.** Por primera vez se podía «tocar» el QL y por primera vez se escuchaban fechas: ¡para últimos de noviembre! Mucho nos tememos que al igual que ocurrió en Inglaterra, la espera será un poco más larga, pero por el momento se pudo ver de cerca el controvertido ordenador y los cuatro paquetes de aplicaciones que lo acompañan. Pero no estaba por ello olvidado nuestro querido Spectrum. Junto a él aparecían unas unidades de discos fabricados por **Sinclair** en Portugal, de apariencia bastante robusta, sobre las que nadie se atrevía aventurar precios o fechas de aparición. Por fin pudimos ver una auténtica unidad de discos para el Spectrum de 220



Por fin llegó la esperada unidad de discos de la mano de Sinclair.



Comcon, *Joystick* programable para cuatro posiciones. Tan sencillo que parece imposible que nadie lo inventase antes.



El rancho. Programas educativos para los más pequeños.

```
CONVERSION DE SISTEMAS DE MEDIDA
 1  METROS A PIES
 2   "  "   PULGADAS
 3   "  "   YARDAS
 4  METROS ↑2 A PIES ↑2
 5   "  "        YARDAS ↑2
 6  MILIMETROS A PIES
 7    "  "     A PULGADAS
 8    "  "     A MILS
 9    "  "     ↑2 A PULGADAS ↑2
10  KILOMETROS A MILLAS NAU.
11    "  "       MILLAS EST.
12  KILOMETROS ↑2 A ACRES
13  CENTIMETROS A PIES
14    "  "        PULGADAS
15  CENTIMETROS ↑2 A PULGADAS ↑2
16    "  "     ↑3 A PIES ↑3
17  GRADOS-RADIANES,18 RAD.-GRAD
 Z  PARA MENU PRINCIPAL
DIGITE SU ELECCION
"C"
```

Matemáticas para todos con el programa Calc.

Kbytes por disco, (160 por cara). Junto a ellos aparecía «un amigo muy personal» según rezaba la publicidad sobre *Verbot*, el robot apenas superior a la altura de un balón de futbol que obedece a la voz.

No menos espectacular resultaba el *joystick* por infrarrojos presentado por **Sitelsa** que también presentaba el *cassette Euromatic*, especial para ordenadores y el conocido *Sweet Talker*, una modalidad de sintetizador de voz. El *cassette* permite la lectura/grabación sin desconectar las conexiones EAR/MIC, incorporando un led indicativo de conexión a la red y otro indicativo de transferencia de datos.

Tics, empresa especializada en el campo de la medición, presentó un controlador que permitía la toma de datos externos a través del *bus* de expansión (en el próximo número de TODOSPECTRUM su inventor nos explicará en detalle este dispositivo).

Tampoco faltó la atractiva tableta digitalizadora *Grafpad* importada por **Ventamatic,** junto con un curioso *joystick* de fácil programación para cualquier tecla y de un diseño un tanto arcaico: el *Comcon*. (Basta colocar las conexiones en las teclas que utiliza el juego para poder utilizar el *joystick*). Pensando en los amantes de la música, presentaban también el *interface Midi* para instrumentos musicales electrónicos, utilizado corrientemente para la conexión a sintetizadores y cajas de ritmos programable.

En el capítulo bibliográfico la firma **Ventamatic** presentó dos libros con dos títulos sugestivos: «Los superjuegos del ZX-Spectrum 16K/48K, y ZX-*Interface* 1 y ZX-*Microdrives:* Qué son, para qué sirven, y cómo se usan.»

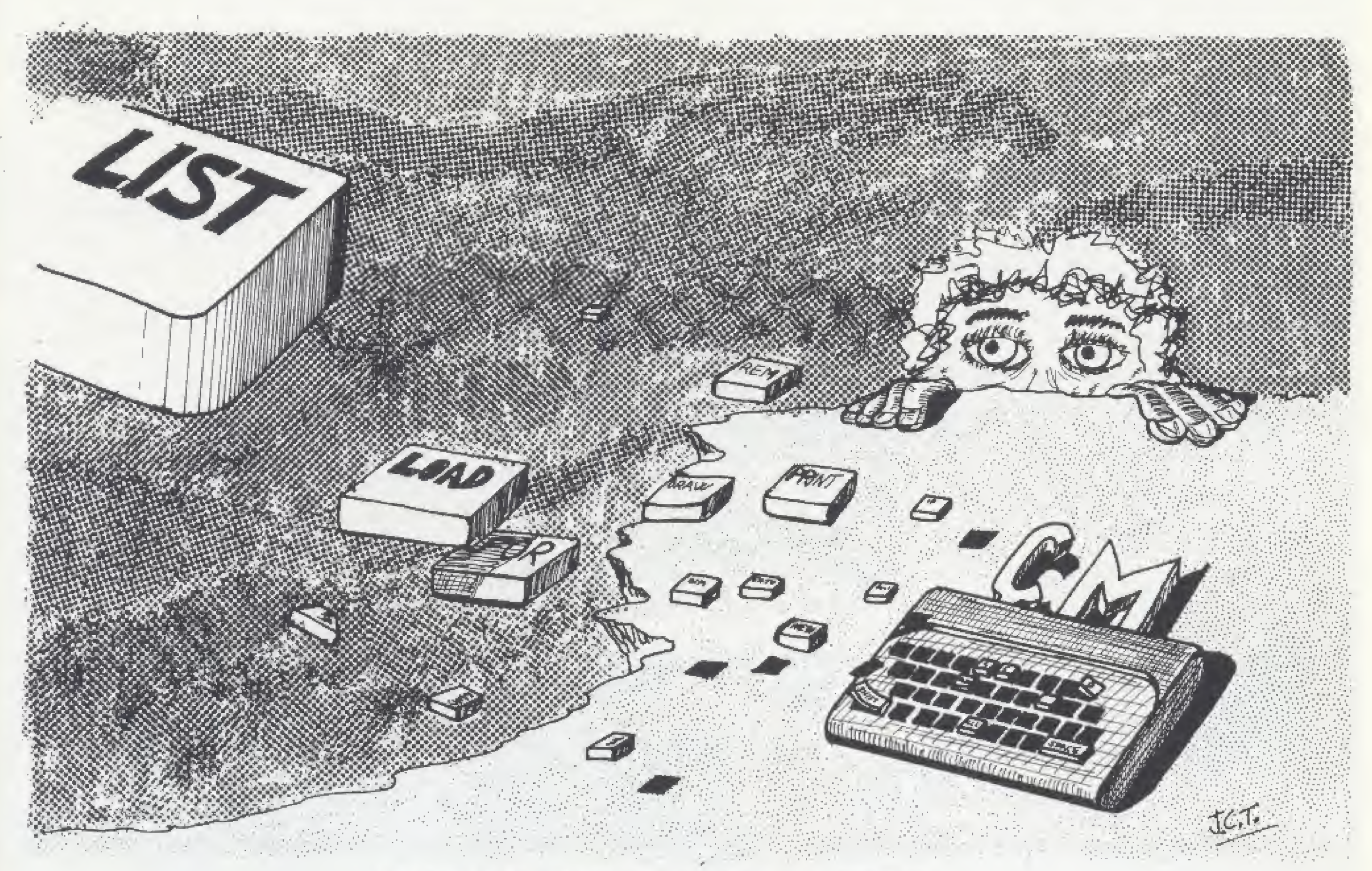
Finalmente indicar que cuantos hacemos TODOSPECTRUM, también estuvimos presentes, al igual que estaremos en el SIMO, en el presente mes de noviembre en Madrid, a fin de daros toda la información que necesitéis y, por supuesto, para conocernos mejor. Para aquellos que no podáis asistir a esta feria, os daremos cumplida información en el número del próximo mes.



Stands especiales para que todos pudiesen «ver y tocar».



Los chicos fueron los grandes protagonistas.



TRES NUEVOS COMANDOS
CON CODIGO MAQUINA

Basic ampliado

En este artículo veremos cómo añadir al Basic tres nuevas instrucciones:
— SCREEN$ n... que resulta equivalente a las siguientes instrucciones Basic: BORDER n : PAPER n : INK 9 : FLASH 0 : BRIGHT 0 : CLS.
— INVERSE... que intercambia los colores de INK y PAPER en la zona de atributos de la pantalla.
— BIN... que ahora significa "Bit INverse" e invierte la pantalla normal.

El INTERFACE I, aparte de permitir un almacenamiento rápido de información en *microdrives*, la conexión vía RS232 y el trabajo en red *(network)*, también nos ofrece la oculta posibilidad de ampliar el BASIC con nuevas instrucciones.

El *interface* permite saltar a una subrutina escrita por el usuario cuando se comete un error sintáctico. Esto significa que se puede llamar subrutinas en lenguaje máquina cambiando la sintaxis de la mayoría de las instrucciones.

Cuando se comete un error sintáctico el control se transfiere a la ROM del *interface*, que llamare-

mos a partir de ahora "shadow ROM" (shadow = sombra), que se encarga de comprobar si el error está producido por una instrucción del *interface*. De ser así, esta se ejecuta sin más complicaciones. En caso contrario, se ejecuta un salto a la dirección que indica la variable del sistema VECTOR.

Esta variable normalmente tiene el valor #01FO (# significa hexadecimal) que produce la generación del mensaje de *syntax error*. Pero podemos cambiar este valor para que indique el comienzo de nuestra subrutina situada en RAM.

La subrutina RAM debe tener dos módulos diferenciados, uno para comprobar la sintaxis de la instrucción modificada y otra para ejecutar realmente dicha instrucción. Al escribir esta parte del programa debe tenerse en cuenta dos cosas:



— Se ha producido el paginado de la ROM, por tanto los RST tienen otro efecto.

— No se produce la lectura de teclado ni el incremento de *frames*.

El listado realizado en lenguaje máquina añade y ejecuta las instrucciones arriba descritas. Está realizado con el ensamblador GENS 3M de HISOFT.

En la primera parte, se crean las variables del sistema mediante el *hook code* (código gancho, o código de transferencia) #31 y a continuación se da el valor del comienzo de nuestra subrutina a la variable VECTOR, acto seguido se devuelve el control al BASIC.

Cada vez que ocurra un error sintáctico, será ejecutada nuestra subrutina. En el punto de entrada de la subrutina el acumulador contendrá el código del primer carácter de la instrucción menos 206. Lo primero que haremos será comprobar si pertenece a alguna de las instrucciones que hemos añadido. De ser así pasaremos el control al analizador sintáctico principal correspondiente. En caso contrario

```
   1 REM ***********************
   2 REM * BASIC AMPLIADO para *
   3 REM *     INTERFACE 1     *
   4 REM ***********************
   5 REM
   6 REM
   7 REM Se baja el RAMTOP
  10 CLEAR 59999
  20  LET SC=0: REM suma de
                  comprobacion
  30 LET START=6E4: REM
          direccion de comienzo
  40 CLS : PRINT "ESPERA, estoy
pokeando el codigo"
  50 FOR i=START  TO START+119
  60 READ A: POKE i,A
  70 LET SC=SC+A
  80 NEXT i
  85 REM
  90 IF SC<>12627 THEN  CLS : PR
INT "ERROR en la suma de comprob
acion": PRINT : PRINT "LO SIENTO
": STOP
 100 REM Escribe la nueva
         sintaxis
 110 CLS : PRINT "El BASIC tiene
 ahora las          siguientes nue
vas instrucciones:"
 120 PRINT TAB 4;"SCREEN$ n";: F
OR i=1 TO 9: PRINT CHR$ 8;: NEXT
 i: PRINT  OVER 1;"_________"
 130 PRINT "Es eqivalente a BORD
DER n:PAPER n:INK 9:BRIGHT 0:FLA
SH 0:CLS"'''
 140 PRINT TAB 4;"INVERSE";: FOR
 I=1 TO 7: PRINT CHR$ 8;: NEXT I
: PRINT  OVER 1;"_______"
 150 PRINT "Cambia los valores d
e INK y      PAPER en la zona de
atributos"'''
 160 PRINT TAB 4;"BIN";: FOR I=1
 TO 3: PRINT CHR$ 8;: NEXT I: PR
INT  OVER 1;"___"
 170 PRINT "Invierte la situacio
n de los     pixels de pantalla"
 180  PRINT #1;"Pulsa cualquier
tecla para        continuar."
 190  PAUSE 1: PAUSE 0
 199 REM  Rutina de salvamento
             del codigo
 200  CLS : PRINT "En que sistem
a deseas salvar     el codigo?"
 210 PRINT TAB 10;"1-CASETE": PR
INT TAB 10;"2-MICRODRIVE": PRINT
  TAB 10;"3-NINGUNO"
 220 LET A$=INKEY$: IF A$<"1" OR
```

generaremos un mensaje de *syntax error* saltando a la posición # 01f0 de la *shadow* ROM.

Supongamos que el carácter es el *token* de SCREEN$ y que por tanto hemos saltado a la línea 300 para realizar el análisis sintáctico principal de la instrucción.

Hacemos llamadas a las subrutinas de análisis de la ROM principal utilizando la instrucción RST #10 seguida de la dirección de la subrutina que queremos ejecutar (conviene hacer notar que hemos paginado a la *shadow* ROM y que por tanto RST #10 no significa escribir un carácter). Hacemos llamadas a NEXTCHR (#0020) para incrementar el puntero de carácter y a NUMEX (#1C82) para evaluar una expresión numérica.

Esta última subrutina comprueba la sintaxis de una expresión numérica y en el momento de la ejecución la calcula y sitúa su resultado en el *stack* (pila) del calculador. También se efectúa una llamada a la subrutina de la *shadow* ROM STEND (#05B7) para comprobar si se ha alcanzado el final de la instrucción.

Los otros dos analizadores sintácticos principales actúan de igual manera. Las rutinas son las mismas, y para ejecutar cualquier otra porción de código haremos el salto mediante RST #10. El retorno se realiza mediante un JP #05C1 (ojo, nada de RET). Mediante este JP devolveremos el control al analizador de BASIC.

Rutinas de ejecución de SCREEN$ n:

El valor de n se toma del stack y se guarda en el registro H. El color del borde se cambia llamando la rutina apropiada de la ROM principal. El color de la tinta (INK) se cambia a blanco o a negro para lograr el máximo contraste, y se guarda junto con el color del papel en la variable del sistema ATTR-P.

Rutinas de ejecución para INVERSE:

Cargamos en BC el comienzo de la zona de atributos de pantalla, y utilizamos los registros A y D para intercambiar los colores de papel y tinta (INK). Posteriormente se incrementa el par de registros BC y se cierra el bucle, que se ejecutará hasta que hayamos cubierto toda la zona de atributos.

Rutinas de ejcución para BIN:

BC se carga con la dirección del comienzo de la memoria de panta-

```
 A$>"3" THEN  GO TO 220
 225 IF A$="3" THEN  GO TO 240
 230 GO SUB 1000*VAL A$
 240  CLS : PRINT "El BASIC AMP
LIADO                  ya es operat
ivo."
 250  RANDOMIZE USR 6e4 : REM ej
ecucion
 260  GO TO 16000 : REM  fin de
programa
 499 REM DATAS del codigo maquin
a
 500 DATA 207,49,1,106,234,237,6
7,183,92,201,254,220,40,11,254,1
5,40,42,254,246,40,76,195
 510 DATA 240,1,215,32,0,215,130
,28,205,183,5,215,148,30,103,215
,151,34,124,7,7,7,254,25,48
 520 DATA 2,246,7,50,141,92,215,
107,13,195,193,5,215,32,0,205,18
3,5,1,0,88,10,230,7,7,7
 530 DATA 7,87,10,230,56,15,15,1
5,130,87,10,230,192,130,2,3,120,
254,91,32,230,195,193,5,215
 540 DATA 32,0,205,183,5,1,0,64,
10,238,255,2,3,120,254,88,32,246
,195,193,5
```

```
 999 REM Rutina de grabacion en
casete
1000 CLS : PRINT "INTRODUCE UNA
CINTA      y         DESCONECTA 'EA
R'"
1010 SAVE "BASICamp"CODE START,1
20
1020 CLS : PRINT "Ya he grabado
los datos en el   casete con el
nombre BASICamp."
1030  RETURN
1999 REM Rutina de grabacion
  en microdrive
2000 CLS : PRINT "Grabando en el
 microdrive. "
2010 INPUT "NOMBRE DEL FICHERO";
F$,"NUMERO DE DRIVE";D
2020 IF F$="" OR LEN F$>10 THEN
 GO TO 2010
2030  PRINT ''"PULSA CUALQUIER T
ECLA PARA       GRABAR ";F$: PRIN
T " EN EL DRIVE NUMERO ";D
2040 PAUSE 1: PAUSE 0
2050 SAVE *"M";D;F$CODE START,12
0
2060 RETURN
```



lla. El acumulador se utiliza para hacer la inversión de la pantalla mediante una operación O EXCLUSIVA con #FF. Aquí también se cierra un bucle que se repite hasta que se ha realizado la inversión de toda la pantalla.

El programa en BASIC es para aquellos que no posean un ensamblador. Introdúzcase el programa y sálvese en *cassette o microdrive* antes de ejecutarlo. Posteriormente ejecútese. Si se produce un mensaje de error en la suma de comprobación *(checksum),* revísense los datos de las líneas 500 a 540. Una vez que todo esté correcto aparecerán en pantalla las nuevas instrucciones y las acciones que estas realizan. Posteriormente se debe optar entre guardar el código máquina en *cassette o microdrive.* Síganse las instrucciones que aparezcan en pantalla. Para utilizar las nuevas instrucciones, una vez salvadas, introdúzcanse las siguientes órdenes directas:

CLEAR 59999: LOAD" "CODE (para *cassette*)
(LOAD "M";d;"nombre"CODE para *microdrive*): RANDOMIZE USR 6e4

Las instrucciones del BASIC ampliado no funcionarán después de un NEW, pero volverán a ser váli-

```
                  10 ;
                  20 ;
                  30 ; CAMBIO DE ATRIBUTOS
                  40 ; PANTALLA CON SCREEN$ n
                  50 ; E INVERSION CON INVERSE
                  60 ; Y BIN PARA USUARIOS DEL
                  70 ; INTERFACE I
                  80 ;
                  90 ;
                 100 ;
EA60             110 INICIA  ORG   60000 ;DIREC DE COMIENZO
EA60  CF         120         RST   #8 ;FIJAR NUEVAS VARIABLES
EA61  31         130         DEFB  #31
EA62  016AEA     140         LD    BC,START
EA65  ED43B75C   150         LD    (VECTOR),BC
EA69  C9         160         RET   ;VUELTA AL BASIC
                 170 ;
                 180 ; COMPROBACION DE SINTAXIS
                 190 ;
EA6A  FEDC       200 START   CP    # DC ;TOKEN DE SCREEN$-206
EA6C  CA7CEA     210         JP    Z,SCRSIN
EA6F  FEOF       220         CP    # OF ;TOKEN DE INVERSE-206
EA71  CA9FEA     230         JP    Z,INVSIN
EA74  FEF6       240         CP    # F6 ;TOKEN DE BIN-206
EA76  CAC5EA     250         JP    Z,BINSIN
EA79  C3F001     260         JP    ERROR
                 270 ;
                 280 ; ANALIZADOR DE SCREEN$
                 290 ;
EA7C  D7         300 SCRSIN  RST   # 10 ;COGE EL SIGUIENTE CARACTER
EA7D  2000       310         DEFW  NEXTCH
EA7F  07         320         RST   # 10 ;COGE EXPRESION NUMERICA
EA80  821C       330         DEFW  NUMEXP
EA82  CDB705     340         CALL  STEND ;COMPRUEBA FINAL INS-
                                   TRUCCION
                 350 ;
                 360 ; RUTINAS DE EJECUCION DE SCREEN$
                 370 ;
EA85  D7         380         RST   # 10 ;PONE EL COLOR EN EL ACUMU-
                                   LADOR
EA86  941E       390         DEFW  UNSTA
EA88  67         400         LD    H,A
EA89  D7         410         RST   # 10 ;CAMBIA BORDER
EA8A  9722       420         DEFW  BORDER
EA8C  7C         430         LD    A,H ;RECUPERA A
EA8D  07         440         RLCA  ;PASA COLOR A LOS BITS 3-5
EA8E  07         450         RLCA
EA8F  07         460         RLCA
EA90  FE19       470         CP    25 ;COMPARA NIVEL DE CONTRASTE
EA92  3002       480         JR    NC,INKO ;TINTA (INK) NEGRA
EA94  F607       490         OR    7 ;TINTA (INK) BLANCA
EA96  328D5C     500 INKO    LD    (ATTR_P),A ;CAMBIO INK Y PAPER
EA99  D7         510         RST   # 10 ;BORRA PANTALLA
EA9A  6BOD       520         DEFW  CLS
EA9C  C3C105     530         JP    BASIC ;VUELTA AL BASIC
                 540 ;
                 550 ;
                 560 ;
                 570 ; ANALIZADOR DE INVERSE
                 580 ;
EA9F  D7         590 INVSIN  RST   # 10 ;COGE SIGUIENTE CARACTER
EAA0  2000       600         DEFW  NEXTCH
EAA2  CDB705     610         CALL  STEND ;COMPRUEBA FIN DE INS-
                                   TRUCCION
                 620 ;
                 630 ; RUTINAS DE EJECUCION DE INVERSE
                 640 ;
EAA5  010058     650         LD    BC,ATR ;BC=COMIENZO ZONA ATRI-
                                   BUTOS
EAA8  0A         660 BUCLE   LD    A,(BC) ;A=ATRIBUTO
EAA9  E607       670         AND   7 ;A=TINTA (INK) ANTIGUA
```

```
EAAB  07       680            RLCA
EAAC  07       690            RLCA
EAAD  07       700            RLCA   ;A=NUEVO PAPEL
EAAE  57       710            LD     D,A ;D=NUEVO PAPEL
EAAF  0A       720            LD     A,(BC);A=ATRIBUTO
EAB0  E638     730            AND    %00111000 ;A=PAPEL ANTIGUO
EAB2  0F       740            RRCA
EAB3  0F       750            RRCA
EAB4  0F       760            RRCA   ;A=NUEVA TINTA (INK)
EAB5  82       770            ADD    A,D ;A=NUEVOS TINTA (INK) Y PAPEL
EAB6  57       780            LD     D,A ;D=NUEVOS TINTA (INK) Y PAPEL
EAB7  0A       790            LD     A,(BC);A=ATRIBUTO
EAB8  E6C0     800            AND    %11000000  ;A=BRILLO (BRIGHT) Y
                                     FLASH
EABA  82       810            ADD    A,D ;A=NUEVO ATRIBUTO
EABB  02       820            LD     (BC),A ;CAMBIO EN PANTALLA
EABC  03       830            INC    BC
EABD  78       840            LD     A,B
EABE  FE5B     850            CP     # 5B ;COMPRUEBA FIN ZONA ATRIBU-
                                     TOS
EAC0  20E6     860            JR     NZ,BUCLE ;CIERRA EL BUCLE
EAC2  C3C105   870            JP     BASIC ;VUELTA AL BASIC
               880   ;
               890   ;
               900   ;
               910   ; ANALIZADOR DE BIN
               920   ;
EAC5  D7       930   BISIN    RST    # 10 ;COGE SIGUIENTE CARACTER
EAC6  2000     940            DEFW   NEXTCH
EAC8  CDB705   950            CALL   STEND ;COMPRUEBA FIN INSTRUC-
                                     CION
               960   ;
               970   ; RUTINAS DE EJECUCION DE BIN
               980   ;
EACB  010040   990            LD     BC,PANTA ;BC=COMIENZO MEMORIA
                                     PANTALLA
EACE  0A       1000  LAZO     LD     A,(BC);A=BYTE DE PANTALLA
EACF  EEFF     1010           XOR    # FF ;SE INVIERTE
EAD1  02       1020           LD     (BC),A ;SE DEVUELVE CAMBIADO
EAD2  03       1030           INC    BC ;SE INCREMENTA BC
EAD3  78       1040           LD     A,B
EAD4  FE58     1050           CP     # 58 ;COMPRUEBA FINAL DE PANTA-
                                     LLA
EAD6  20F6     1060           JR     NZ,LAZO ;CIERRA EL BUCLE
EAD8  C3C105   1070  ;        JP     BASIC ;VUELTA AL BASIC
               1080  ;
               1090  ;
               1100  ;
               1110  ;
               1120  ; ETIQUETAS
               1130  ;
5CB7           1140  VECTOR   EQU    # 5CB7 ;VECTOR PARA AMPLIACION
                                            BASIC
01F0           1150  ERROR    EQU    # 01F0 ;ERROR SINTACTICO (SHA-
                                            DOW)
0020           1160  NEXTCH   EQU    # 0020 ;COGE SIGUIENTE CARACTER
                                            (PRINC)
1C82           1170  NUMEXP   EQU    # 1C82 ;COGE EXPRESION NUMERI-
                                            CA (PRINC)
05B7           1180  STEND    EQU    #05B7 ;COMPRUEBA FIN INSTRUC.
                                            (SHADOW)
1E94           1190  UNSTA    EQU    # 1E94 ;PASA DE STACK A ACUM.
                                            (PRINC)
2297           1200  BORDER   EQU    # 2297 ;CAMBIA COLOR DEL BORDE
                                            (PRINC)
5C8D           1210  ATT_P    EQU    # 5C8D ;VARIABLE DE ATRIBUTOS
0D6B           1220  CLS      EQU    #0D6B ;BORRA PANTALLA (PRINC)
05C1           1230  BASIC    EQU    #05C1 ;VUELTA AL BASIC (SHADOW)
5800           1240  ATR      EQU    # 5800 ;COMIENZO ZONA DE ATRIBU-
                                            TOS
4000           1250  PANTA    EQU    # 4000 ;COMIEZO MEMORIA DE PAN-
                                            TALLA
               1270;
```

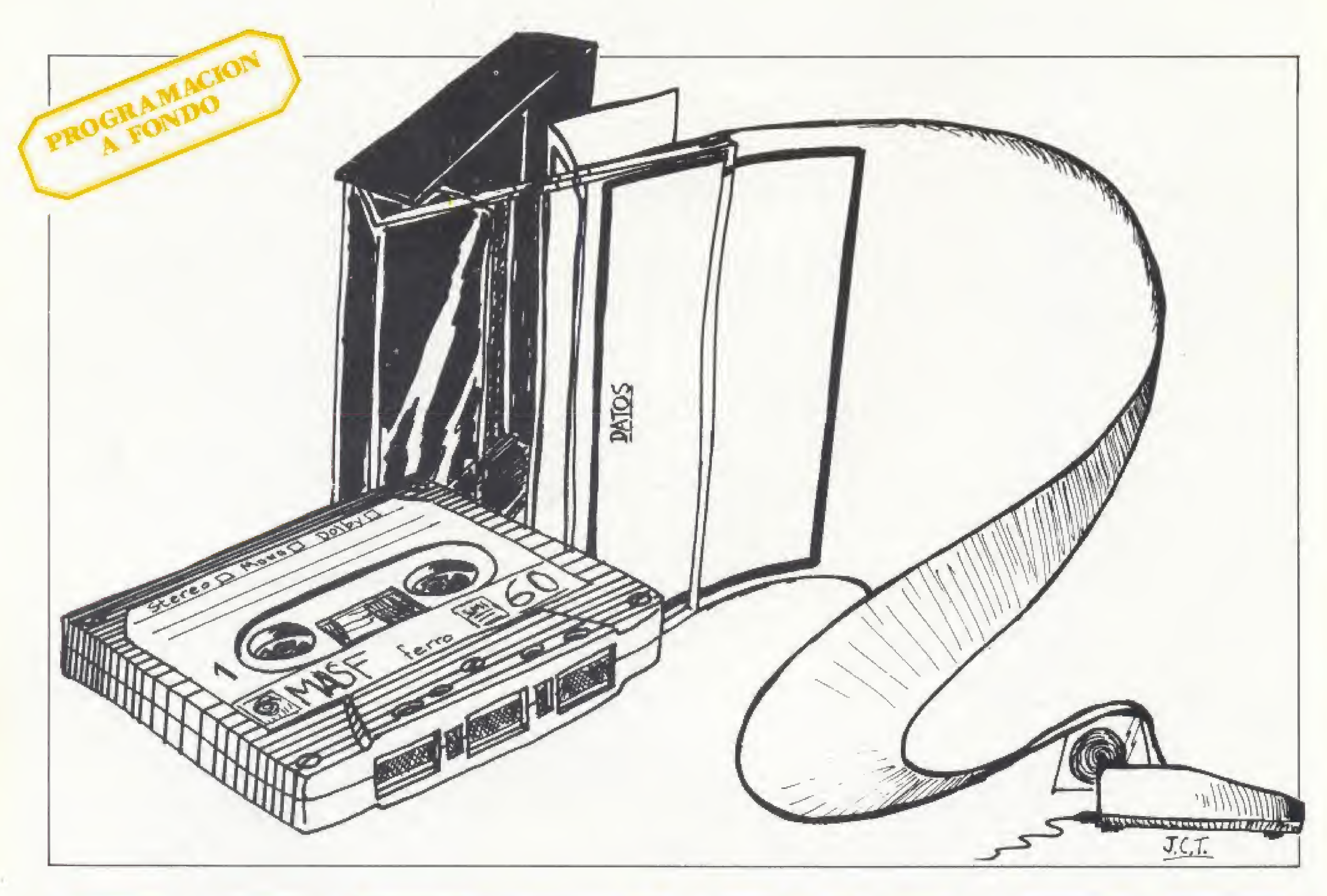
das si se ejecutan un RANDOMIZE USR 6e4.

Una vez que estén las nuevas instrucciones operativas, escriba SCREEN$ 1, y toda la pantalla se pondrá de color azul, así como blancas las teclas. Las instrucciones INVERSE y BIN tendrán el mismo efecto visual en la pantalla. La instrucción BIN influye en los resultados de la función POINT ya que si un punto estaba encendido, ahora estará apagado, y viceversa. Una aplicación específica de BIN es la inversión de volcados de pantalla.

#1c82	**principal**	**tomar expresión numérica**
#1c8c	**principal**	**tomar expresión alfanumérica**
#0018	**principal**	**pasa el *byte* actual al acumulador**
#0020	**principal**	**pasa el *byte* siguiente al actual acumulador**
#01f0	*shadow*	**error sintáctico**
#05b7	*shadow*	**comprobación de fin de instrucción**
#05c1	*shadow*	**retorno al intérprete BASIC**
#0020	*shadow*	**produce un error de la *shadow* ROM (debe ir seguido por un número comprendido entre #16 y #ff)**
#0028	*shadow*	**produce un error de la ROM principal (debe cargarse el código de error en la posición #5c3a)**

¿Le pareció complicado? Si tiene alguna duda no lo dude: escríbanos. Nuestro «experto» en código máquina promete contestar a todos.

Anímese, no se conforme con las funciones que incorpora su Spectrum y crease aquellas que sustituyan a esas viejas rutinas que viene utilizando desde hace tiempo. Si le funciona, nuestra más ferviente enhorabuena, y si lo coloca en un sobre a nuestra dirección, doble enhorabuena por la publicación de su futuro artículo.



SOFTWARE para ordenar el SOFTWARE

Para realizar un buen programa hay que olvidar el ordenador. Al menos mientras no se tenga claro lo que se quiere hacer y cómo se va a llevar a cabo. Eso es lo que hicimos con el programa que le mostramos a continuación y que le servirá para ordenar sus programas almacenados en cassette de una forma tan original como didáctica.

Problema

Realizar un programa que almacene los datos relativos a los programas que se guarda en cada *cassette* y obtener una copia por impresora. Deberá permitir la actualización de la información conforme se vaya cambiando los datos del *cassette*. Aprovechando que la impresora de Sinclair o cualesquiera otra similar, presenta un ancho de papel igual al ancho del *cassette*, también deberá realizar una copia por impresora para adaptarse al *cassette*.

Analizando el problema en profundidad, el programa debe posibilitar la entrada de datos por teclado. Las opciones de impresión por pantalla e impresora serán opcionales y según la información selec-

cionada. Aprovechando la posibilidad del Spectrum de grabar los datos conjuntamente con el programa, se grabará la información utilizando este procedimiento, lo que lleva a proteger el programa ante el posible uso de la instrucción RUN. Por tanto, otra de las opciones que habrá de tener el programa será la anulación de la información que se estime conveniente. Para programar de forma ordenada, las distintas opciones se incorporarán en forma de "menu", que constituirá la parte principal desde donde se bifurcará a los módulos o bloques de instrucciones que realicen las distintas opciones.

Procedimiento a seguir

1. Inicializar las variables a utilizar (variables numéricas, alfanuméricas y tablas).

2. Presentar opciones del menú:

— Introducción de los datos (creación del archivo).

— Impresión de los datos (pantalla/impresora).

— Actualización de los datos.

— Eliminación de los datos.

— Impresión en formato especial para el *cassette*.

— Grabación de los datos.

— Eliminación de todos los datos.

Finalizada la opción elegida se volverá al menú para nuevas opciones.

3. La creación del archivo supone:

— Definir un contador de archivos (el número de archivos existentes + 1).

— Introducción de datos.

— Incrementar el contador de archivos.

— Sub-opciones para poder visualizar los datos introducidos, crear un nuevo archivo o volver al menú.

4. Operaciones comunes a la impresión de datos por pantalla o impresora serán:

— Introducción del código de archivo.

— Impresión del contenido en pantalla.

— Sub-opciones: impresión en impresora (COPY), visualización de otro archivo o volver al menú.

5. La revisión de datos deberá tener los siguientes pasos:

— Introducción del código de archivo a revisar.

— Impresión por pantalla de la información del archivo solicitado.

— Introducción de nuevos datos si así se desea.

— Visualización de los nuevos datos introducidos, en su caso.

— Volver al menú.

6. Para eliminar un archivo:

— Introducción del código de archivo a eliminar.

— Visualizar el nombre y referencia del archivo que se pretende eliminar.

— Confirmar la eliminación.

— Borrar el contenido del archivo seleccionado.

— Trasladar los datos de archivos por encima del eliminado, si los hubiere, para mantener la secuencia de numeración automática de archivos.

— Actualizar el contador de archivos (número de archivos existentes − 1).

— Sub-opción: Listado de los archivos (COPY) para tener una copia de la nueva situación y volver al menú.

— Volver al menú.

```
   5 LET S$="FICHERO*1"
  10 REM FICHERO CASSETTE
  20 DIM C$(10,4)
  30 DIM N$(10,15)
  40 DIM P$(10,10,15)
  50 DIM T(10,10,2)
  60 LET X=0
  70 LET F=0
  80 LET E$="
                 "
  90 LET A$="REF. CINTA:   "
 100 LET B$="NOMBRE CINTA:"
 110 LET F$="FICHERO:  "
 120 LET G$="NOMBRE PROG.:"
 130 LET H$="CONTADOR:"
 140 LET I$="DESDE  "
 150 LET TF=0
 160 LET X$="
            "
 170 LET Y$="
            "
 180 LET Z$="
            "
 190 REM ********
 191 REM **MENU**
 192 REM ********
 200 PRINT S$
 210 PRINT TAB 10;"MENU"
 220 PRINT
 230 PRINT "1. CREAR FICHERO"
 240 PRINT ,,"2. VER/LISTAR FICH
ERO"
 250 PRINT ,,"3. COMPROBAR FICHE
RO"
 260 PRINT ,,"4. BORRAR FICHERO"
 270 PRINT ,,"5. LISTAR ETIQUETA
S DE CINTA"
 280 PRINT ,,"6. GUARDAR FICHERO
S"
 290 PRINT ,,"7. BORRAR TODOS LO
S FICHEROS"
 300 PRINT ,,"INTRODUZCA LA OPCI
ON"
 310 INPUT M
 320 IF M<1 OR M>7 THEN GO TO 30
0
 325 IF M=6 OR M=7 THEN GO TO M*
1000
 330 CLS
 340 GO SUB M*1000
 350 CLS
 355 IF X=1 THEN GO TO 20
 360 GO TO 210
 900 REM *******************
 910 REM ** CREAR FICHERO **
 920 REM *******************
1000 LET F=TF+1
1010 IF F<=10 THEN GO TO 1020
1015 PRINT "*NO HAY SITIO PARA O
TRO FICHERO*"'"AL VOLVER AL MENU
 ELIGA LA       "'"OPCION 'GUARDA
R FICHERO' PARA   "'"ALMACENAR L
OS FICHEROS ACTUALES "'"LUEGO BO
RRE LOS FICHEROS PARA    "'"CREAR
 UNA NUEVA TABLA."
1016 PAUSE 300
1017 RETURN
1020 PRINT "TRABAJANDO EN FICHER
O NO.";F
1030 PRINT AT 2,0;"TECLEE REF. C
ASSETTE (4 CARAC.)"
1040 INPUT C$(F)
1050 PRINT AT 2,0;E$;AT 2,0;B$;"
?(MAX.15 CAR.)"
```



7. Impresión de los datos en formato de *cassette*.

— Introducción del código de archivo a imprimir.

— Impresión de la información en el formato de *cassette*.

— Volver al menú.

8. La grabación de los datos supone:

— Posibilidad de que el usuario elija un nombre para la grabación.

— Confirmar la grabación (volver al menú en caso negativo).

— Detener el programa hasta que el *cassette* esté preparado.

— Grabar el programa en forma de autoejecución. (SAVE "LINE" XXXX).

— Volver al menú.

9. Eliminar toda la información:

— Confirmar la eliminación.

— Utilizar CLEAR para mayor rapidez.

— Reinicializar las variables.

— Volver al menú.

```
REF. CINTA:   HIFI   FICHERO: 1
NOMBRE CINTA:  todospectrum

NOMBRE PROG.:              CONTADOR:
                           DESDE

1 Comecocos                0     50
2 The hobbit               60    80
3 Ajedrez                  90    100
4 Graficos                 105   110
5 Easel                    120   150
6 Estratego                160   170

REF. CINTA:   HIFI   FICHERO: 1
NOMBRE CINTA:  todospectrum

7 Galaxia 2000             200   220
8                          0     0
9                          0     0
10                         0     0
```

Formato especial para cassette.

Variables

C$ (10, 4)

Referencia del *cassette* (4 caracteres).

N$ (10, 15)

Nombre del *cassette* (15 caracteres).

P$ (10, 10, 15)

10 nombres de programas (15 caracteres) para cada uno de los 10 *cassettes*.

T (10, 10, 2)

Numeración del contador inicial y final para cada uno de los programas de los distintos *cassettes*.

F

Código del *cassette* con el que se está trabajando.

TF

Contador de archivos (máximo 10).

M$

Variable alfanumérica utilizada en los menús.

M

Variable numérica utilizada en los menús.

S$

Nombre del programa.

N, L

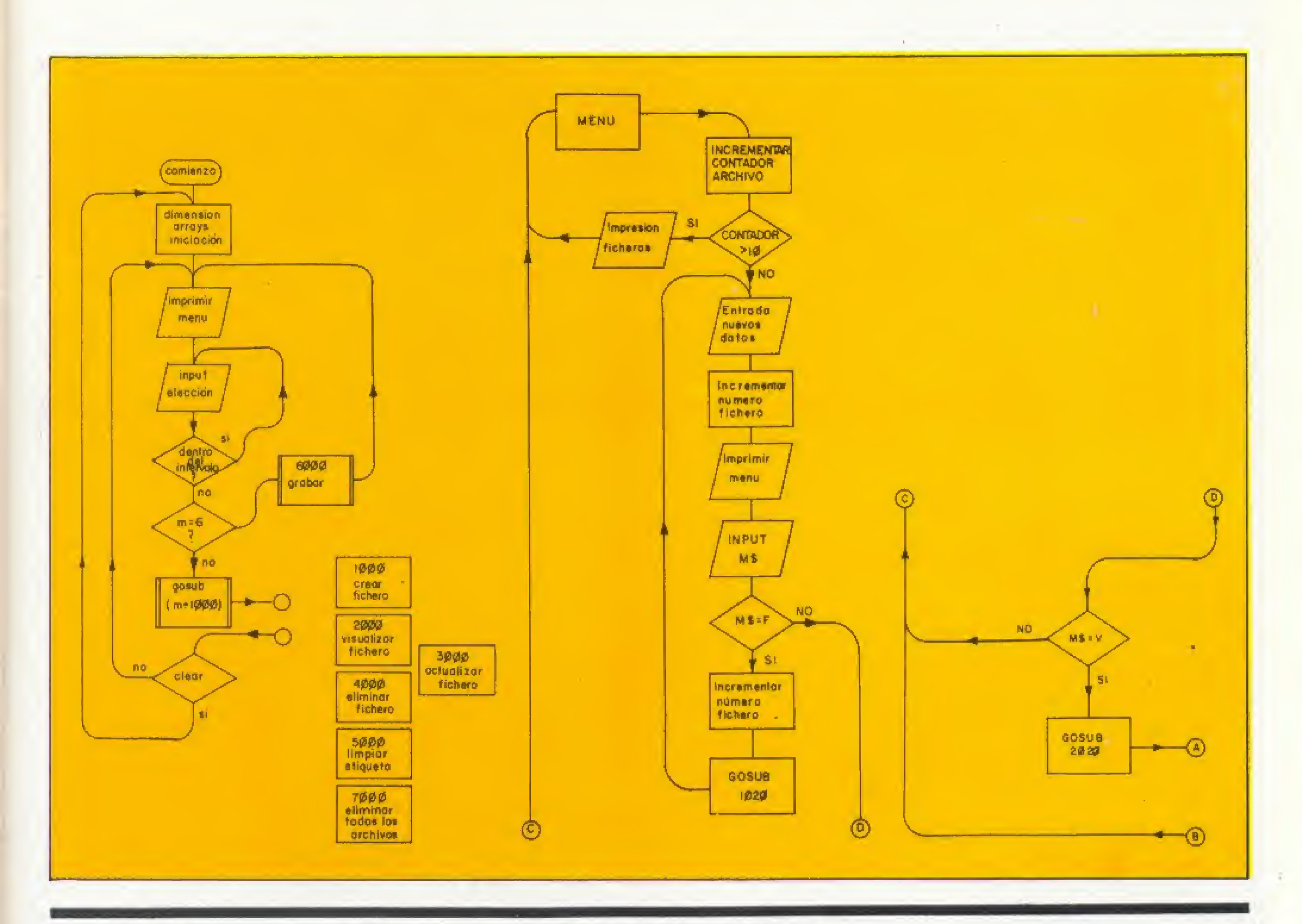
Variables utilizadas en bucles.

E$

Variable nula para utilizar en impresión.

A$, B$, F$,

```
1060 INPUT N$(F)
1070 PRINT AT 2,0;E$;AT 2,0;"TEC
LEE NOMBRE DE PROG. Y CUENTA"
1090 FOR N=1 TO 10
1100 PRINT AT 4,0;"PROGRAMA ";N
1110 PRINT G$;" ?(15 CHRS MAX)"
1120 INPUT P$(F,N)
1130 PRINT AT 5,0;E$;AT 5,0;H$;"
 DESDE?"
1140 INPUT T(F,N,1)
1150 PRINT AT 6,0;"HASTA  ?  "
1160 INPUT T(F,N,2)
1165 CLS
1170 PRINT AT 7,0;E$;AT 5,0;"OTR
O PROGRAMA ?(S/N)"
1180 INPUT M$
1190 IF M$="N" THEN GO TO 1220
1200 IF M$<>"N" AND M$<>"S" THEN
 GO TO 1170
1210 NEXT N
1220 CLS
1225 LET TF=TF+1
1230 PRINT "CREAR OTRO FICHERO (
F)             VER FICHEROS (V)
               MENU (M) ?"'"TECLEE
F,V O M."
1240 INPUT M$
1250 IF M$<>"F" THEN GO TO 1290
1260 LET F=F+1
1270 CLS
1280 GO SUB 1020
1290 CLS
1295 IF M$="V" THEN GO SUB 2020
1300 RETURN
1980 REM *******************
1990 REM *** VER FICHERO ***
1991 REM *******************
2000 PRINT "QUE  FICHERO?"
2010 INPUT F
2015 REM *ENTRADA DESDE OTRAS SU
BRUTINAS*
2020 CLS
2030 IF TF<>0 AND F<=TF AND F>=1
 THEN GO TO 2110
2040 PRINT "NO EXISTE NINGUN FIC
HERO CON ESENUMERO. MENU (M) O V
ER OTRO FI- CHERO (V)."
2060 INPUT M$
2070 IF M$="V" THEN GO TO 2000
2080 IF M$="M" THEN RETURN
2090 PRINT "SIGA LAS INSTRUCCION
ES"
2100 GO TO 2050
2110 CLS
2120 PRINT F$;F
2130 PRINT ,,A$;C$(F)
2140 PRINT "*******************
*************"
2150 PRINT G$;TAB 20;H$;TAB 20;I
$
2160 PRINT "-------------------
-------------"
2170 FOR N=1 TO 10
2180 PRINT N;".";P$(F,N);TAB 20;
T(F,N,1);TAB 26;T(F,N,2)
2185 NEXT N
2190 PRINT AT 21,0;"LISTAR(L),SI
GUIENTE(S),MENU(M)?"
2200 INPUT M$
2210 PRINT AT 21,0;E$
2220 IF M$="L" THEN COPY
2230 IF M$<>"S" THEN GO TO 2310
2240 LET F=F+1
2250 IF F<=TF THEN GO TO 2300
```




```
2255 CLS
2260 PRINT "NO HAY NUM. TAN ALTO
S"
2270 PAUSE 150
2280 CLS
2290 GO TO 2310
2300 GO SUB 2020
2310 RETURN
2980 REM *******************
2990 REM **REVISAR FICHERO**
2995 REM *******************
3000 PRINT "QUE FICHERO REVISO ?
(1 AL 10)"
3010 INPUT F
3020 IF F<=TF THEN GO TO 3060
3030 PRINT "NO EXISTE"
3040 PAUSE 150
3050 RETURN
3060 CLS
3070 PRINT "LAS CARACTERISTICAS
DEL FICHERO ";F;" SE LISTARAN. P
ULSE 'ENTER'"'"SI ESTA CORRECTO
O TECLEE DATOS NUEVOS"
3080 PAUSE 150
3090 POKE 23692,-1
3100 PRINT A$;C$(F)
3110 INPUT M$
3120 IF M$="" THEN GO TO 3170
3130 LET C$(F)=M$
3140 CLS
3150 POKE 23692,-1
3160 PRINT "NUEVO";A$;C$(F)
3170 POKE 23692,-1
3180 PRINT B$;" ";N$(F)
3190 INPUT M$
3200 IF M$="" THEN GO TO 3265
3210 LET N$(F)=M$
3220 CLS
3230 POKE 23692,-1
3240 PRINT A$;C$(F)
3250 POKE 23692,-1
3260 PRINT "NUEVO NOMBRE DEL CAS
SETTE ";N$(F)
3265 POKE 23692,-1
3270 FOR N=1 TO 10
3380 PRINT AT 20,0;"PROG.";N;" N
OMBRE ";P$(F,N)
3390 INPUT M$
3400 IF M$<>"" THEN LET P$(F,N)=
M$
3420 POKE 23692,-1
3430 PRINT AT 21,0;"CONTADOR INI
CIAL:";T(F,N,1);"
3440 INPUT M$
3450 IF M$<>"" THEN LET T(F,N,1)
=VAL M$
3460 PRINT AT 21,0;"DESDE CINTA:
";T(F,N,1);" A LA:";T(F,N,2)
3470 INPUT M$
3480 IF M$<>"" THEN LET T(F,N,2)
=VAL M$
3490 PRINT AT 21,0;"DESDE CINTA:
";T(F,N,1);" A LA:";T(F,N,2)
3495 POKE 23692,-1
3500 NEXT N
3510 RETURN
3990 REM *******************
4000 REM **BORRAR FICHERO**
4001 REM *******************
4010 PRINT "QUE FICHERO DESEA BO
RRAR ?"'"(1 AL 10)"
4020 INPUT F
4030 IF F<=TF AND F>=1 THEN GO T
O 4070
```



G$, H$, I$, X$, Y$, Z$

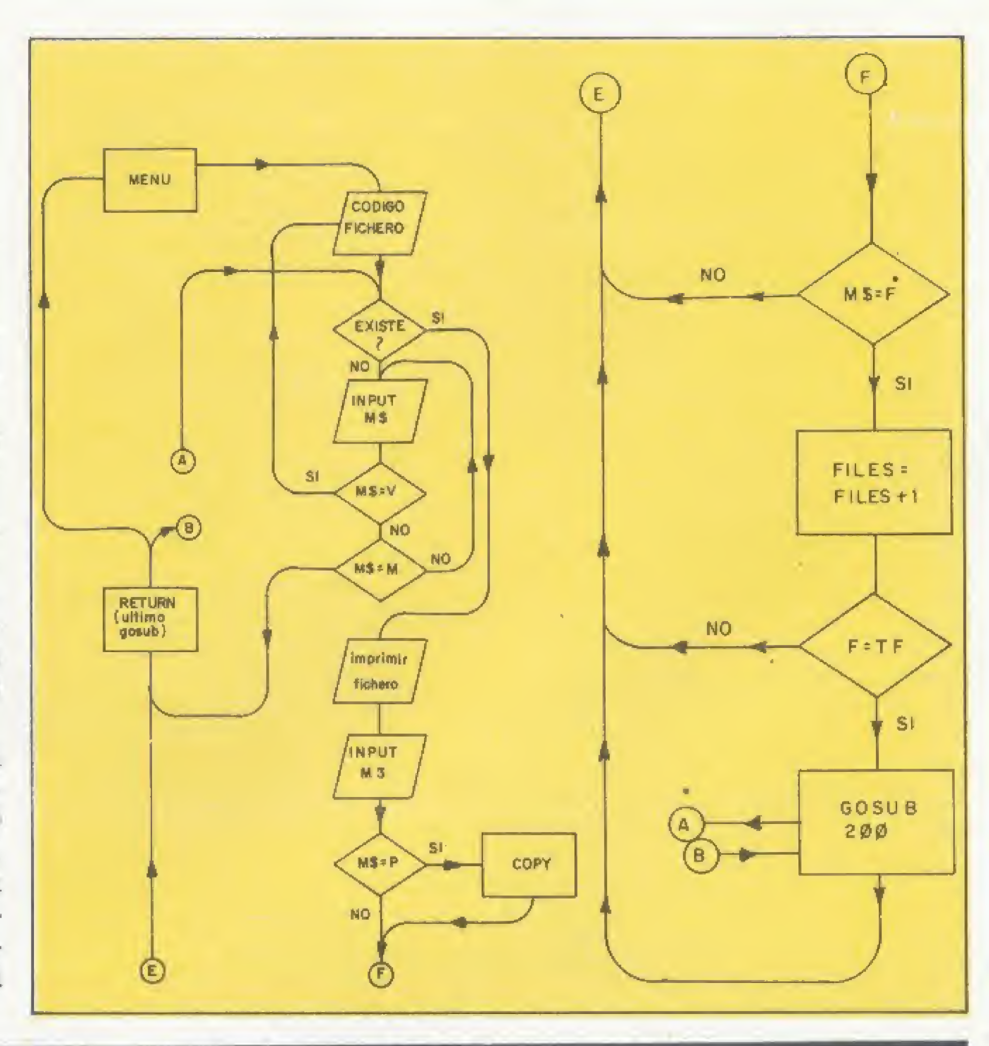
Variables utilizadas para la impresión.

X

Indicador del uso de la instrucción CLEAR para pasar, en su caso, a la reinicialización de variables.

Otra consideración a realizar antes de meterse a realizar el programa es la utilización de los módulos mediante subrutinas. Siempre es aconsejable evitar toda complicación, y ya que siempre se devuelve el control al menú, se evitarán las subrutinas siempre que sea necesario cambiándose por los GOTO.

Inicialmente se preparará para almacenar los nombres de programas con un número máximo de 10 por *cassette* y 10 *cassettes* en total. Posteriormente, según la memoria disponible, se puede cambiar fácilmente para aumentar la capacidad. Un organigrama nos ayudará a ver el proceso de forma clara.




```
4040 PRINT "NO EXISTE ESE FICHER
O"
4050 PAUSE 150
4060 GO TO 4430
4070 PRINT F$;F,,A$;C$(F)
4080 PRINT B$;" ";N$(F)
4090 PRINT AT 10,0;"PULSE LA D P
ARA CONFIRMAR ",,,"O M PARA VER
EL MENU."
4100 INPUT M$
4110 IF M$<>"D" THEN GO TO 4430
4120 PRINT AT 14,0;"TODOS LOS NU
M. DE FICHERO >";F,"SE REDUCIRAN
 EN UNA UNIDAD."
4125 IF F=TF THEN GO TO 4220
4130 FOR N=F TO TF-1
4140 LET N$(N)=N$(N+1)
4150 LET C$(N)=C$(N+1)
4160 FOR L=1 TO 10
4170 LET P$(N,L)=P$(N+1,L)
4180 LET T(N,L,1)=T(N+1,L,1)
4190 LET T(N,L,2)=T(N+1,L,2)
4200 NEXT L
4210 NEXT N
4220 LET C$(TF)=E$
4230 LET N$(TF)=E$
4240 FOR N=1 TO 10
4250 LET P$(TF,N)=E$
4260 LET T(TF,N,1)=0
4270 LET T(TF,N,2)=0
4280 NEXT N
4290 LET TF=TF-1
4300 CLS
4310 PRINT "FICHERO BORRADO"
4320 PRINT ''"NUEVO LISTADO:"
4330 PRINT
4340 FOR N=1 TO TF
4350 PRINT F$;N;" ";A$;C$(N)
4360 NEXT N
4370 PRINT AT 21,0;"COPIAR LISTA
DO NUEVO (C) O VER  EL MENU (M).
"
4380 INPUT M$
4390 IF M$<>"C" THEN GO TO 4430
4400 PRINT AT 21,0;E$
4410 COPY
4420 CLS
4430 RETURN
4990 REM **********************
5000 REM **IMPRIMIR ETIQUETAS**
5010 REM **********************
5020 PRINT "QUE FICHERO QUIERE L
ISTAR COMO  ETIQUETA DE CINTA? (
1 AL 10)"
5030 INPUT F
5040 IF F>=1 AND F<=TF THEN GO T
O 5090
5050 PRINT "*NO EXISTE ESE FICHE
RO*"
5060 PAUSE 150
5070 CLS
5080 GO TO 5280
5090 PRINT ,,"EL FICHERO ";F;" S
E LISTARA."''"COMPRUEBE LA IMPR
ESORA Y PULSE  UNA TECLA PARA EM
PEZAR"
5100 IF INKEY$="" THEN GO TO 510
0
5120 LPRINT X$;"█";A$;C$(F);TAB
20;F$;F;TAB 31;"█"
5130 LPRINT Z$;"█";B$;" ";N$(F);
TAB 31;"█"
5140 LPRINT Z$;X$;"█";G$;TAB 20;
"█";H$;TAB 31;"█"
```

```
            MENU
1. CREAR FICHERO
2. VER/LISTAR FICHERO
3. COMPROBAR FICHERO
4. BORRAR FICHERO
5. LISTAR ETIQUETAS DE CINTA
6. GUARDAR FICHEROS
7. BORRAR TODOS LOS FICHEROS
INTRODUZCA LA OPCION
```

```
FICHERO: 1
REF. CINTA:   HIFI
********************************
NOMBRE PROG.:        CONTADOR:
                     DESDE
--------------------------------
1.Comecocos           0      50
2.The hobbit          60     80
3.Ajedrez             90     100
4.Graficos            105    110
5.Easel               120    150
6.Estratego           160    170
7.Galaxia 2000        200    220
8.                    0      0
9.                    0      0
10.                   0      0

LISTAR(L),SIGUIENTE(S),MENU(M)?
```

Comentarios

No hace falta decir que es un programa un poco largo. Pero su buena estructuración lo hace especialmente aconsejable para seguir de cerca los distintos algoritmos de que se componen sus principales módulos. El programa no es "a prueba de tontos", pero verifica la mayor parte de las respuestas, pidiendo la confirmación de las opciones que pueden suponer la pérdida parcial o total de la información almacenada.

El programa puede mejorarse considerablemente de dos formas. Mediante la incorporación de una rutina de búsqueda para encontrar el *cassette* donde se encuentra un programa determinado. La segunda posibilidad de mejora se refiere a la posibilidad de imprimir la información de varios *cassettes* de forma simultánea en la opción 4, ya que se vuelve al menú después de cada impresión de un *cassette*. Piense sobre los cambios a introducir en las líneas 2190 a 2310.

Si piensa que el programa no es demasiado útil, considere la original idea de utilizar el formato de impresión idéntico al *cassette* para ordenar sus programas, la posibilidad de aumentar la dimensión de las variables y, por supuesto, la modificación del programa para guardar otro tipo de información: libros, discos, etc.

```
5150 LPRINT "▌";TAB 20;"▌";I$;TA
B 31;"▌"
5160 LPRINT X$
5170 FOR N=1 TO 6
5180 LPRINT "▌";N;" ";P$(F,N);TA
B 21;T(F,N,1);TAB 25;T(F,N,2);TA
B 31;"▌";Z$
5190 NEXT N
5200 LPRINT Y$;X$
5210 LPRINT "▌";A$;C$(F);TAB 20;
F$;F;TAB 31;"▌"
5220 LPRINT Z$;"▌";B$;" ";N$(F);
TAB 31;"▌"
5230 LPRINT Y$;X$
5240 FOR N=7 TO 10
5250 LPRINT "▌";N;" ";P$(F,N);TA
B 21;T(F,N,1);TAB 25;T(F,N,2);TA
B 31;"▌";Z$
5260 NEXT N
5270 LPRINT Y$
5280 RETURN
5990 REM ********************
6000 REM **GUARDAR FICHEROS**
6001 REM ********************
6005 CLS
6010 PRINT "RUT. DE ALMACENAMIEN
TO EN CINTA "
6020 PRINT ,,"EL NOMBRE DEL PROG
RAMA ACTUAL ES"
6030 PRINT ,," "" ";S$;" "" "
6040 PRINT ,,"TECLEE EL NUEVO NO
MBRE DEL PRO- GRAMA O PULSE ENTE
R SI LO QUIEREGUARDAR CON EL NOM
BRE ANTIGUO"
6050 PRINT ,,"PULSE M PARA MENU"
6060 INPUT M$
6070 IF M$="M" THEN RETURN
6080 IF M$="" THEN GO TO 9900
6090 LET S$=M$
6100 GO TO 9900
6990 REM ******************
7000 REM **BORRAR FICHEROS**
7005 REM ******************
7010 PRINT "TECLEE D PARA CONFIR
MAR QUE DE- SEA BORRAR TODOS LOS
 FICHEROS"
7020 PRINT ,,"PULE M PARA VOLVER
 AL MENU."
7030 INPUT M$
7040 IF M$<>"D" THEN GO TO 7080
7050 CLEAR
7060 PRINT ,,"NOMBRE PARA ESTE F
ICHERO"
7070 INPUT S$
7080 LET X=1
7090 GO TO 350
9890 REM ******************
9900 REM **AUTO-EJECUCION**
9901 REM ******************
9910 CLS
9920 PRINT "EL NOMBRE DEL PROGRA
MA ES "
9930 PRINT " "" ";S$;" "" "
9940 PRINT "**TOME NOTA**"
9950 PRINT ,,"PREPARE EL CASSETT
E PARA GRABAR Y PULSE CUALQUIER
TECLA."
9960 IF INKEY$="" THEN GO TO 996
0
9970 SAVE S$
9980 CLS
9990 GO TO 200
```





En los últimos meses, el mercado inglés ha lanzado una considerable cantidad de software para aumentar —y en el mayor de los casos facilitar— las posibilidades gráficas del Spectrum. El VU-3D fue uno de los primeros, nacido de la célebre casa **Psion,** *y uno de los más conocidos en España, objeto de estudio de este artículo.*

EL VU-3D es un programa de diseño gráfico que nos permite definir objetos en tres dimensiones sacando una presentación por pantalla o por impresora de una visión del objeto desde el punto de vista que queramos, permitiendo que el usuario se forme una idea general del objeto en tres dimensiones a base de rotarlo y moverlo.

El programa necesita un Spectrum de 48 KRAM, un *cassette* y una televisión. Si se tiene una impresora se pueden sacar copias impresas de los dibujos para mayor gloria del creador.

La carga del programa se realiza con dificultad usando las instrucciones LOAD "" que son habituales en la mayoría de los programas comerciales. Una vez cargado se nos pregunta si queremos cargar un fichero de cinta o si vamos a leer un dibujo anteriormente creado y que hayamos tenido la precaución de salvar con anterioridad.

Después de cargar el fichero (si cargamos alguno) nos aparece un menú con siete opciones, la opción 4, crear figura es la que hace falta para crear un objeto y si no cargamos ningún dibujo de cinta al principio el programa no va al menú principal, sino que se pasa directamente al modo de creación de objetos. En esta rutina debemos crear el objeto que posteriormente se representará en perspectiva.

El modo en que definimos un objeto no está extenso de complejidad y se necesita una buena visión espacial (imaginación en tres dimensiones) para poder crear el dibujo que deseamos. En la parte superior de la pantalla nos aparecen una serie de comandos, mientras el centro se deja para el dibujo de las distintas partes que componen el objeto. Antes de empezar la definición del dibujo debemos dividir nuestro objeto en componentes sencillos. Cada uno de éstos debe ser una figura geométrica sencilla y sin curvas (cubos, pirámides de tres o más lados, etc.) y entre todos deben formar el objeto. El modo de funcionamiento es similar a un juego de construcción de madera

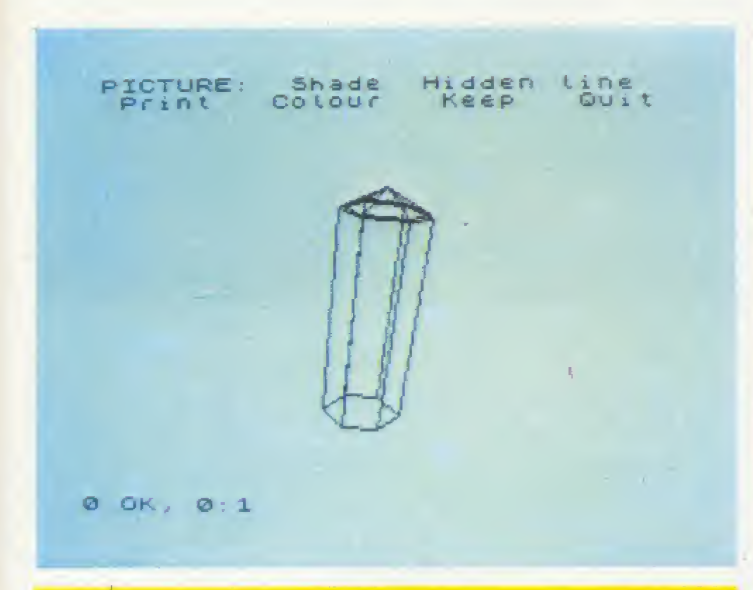


Figura 1

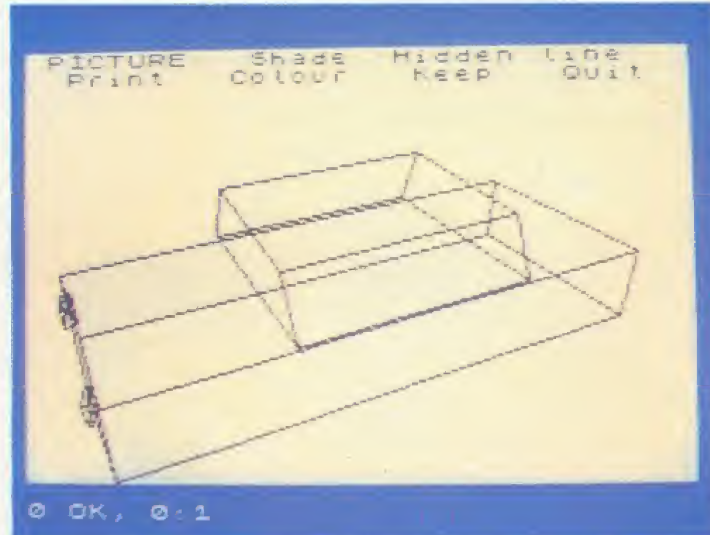


Figura 2

compuesto por las piezas geométricas fundamentales. Para construir un edificio debemos apilar las piezas de un modo lógico. En este programa el funcionamiento es similar pudiéndonos definir las piezas de la forma que queramos (pero dentro de unos límites) y las vamos apilando según las definimos.

La pantalla está dividida en dos ejes X e Y (ancho y largo) y un tercer eje Z destinado a la altura se muestra en sucesivos cortes secuenciales, siendo el primero el 0 (altura del suelo) ascendiendo progresivamente hasta la parte superior del objeto. Por tanto nuestro punto de vista es como si estuviésemos mirando desde el cielo hacia abajo.

Los comandos disponibles son: *Open, Close, Figure, Quit, Next Z, Reduce* y las flechas de movimiento del cursor.

El comando *Next Z* nos permite desplazarnos por las distintas alturas del dibujo, el 0 es el suelo, el 1 es un punto más arriba, el 10 son diez puntos más, etc. Hay que tener en cuenta que una vez que se sube, no se puede bajar. Por lo que debemos tener cuidado al dibujar para no olvidarnos nada, lo que obligaría a repetirlo todo.

Open nos permite definir la parte inferior de la figura en el plano z en que nos encontremos en este momento. Todas las figuras que dibujemos aquí deben ser cerradas y si no las cerramos nosotros, las cerrará el programa al salirnos de la opción. Para dibujar una figura nos colocamos en el punto en que queramos empezar a dibujar, pul-

VU-3D en el cine



Como profesional del mundo de la imagen, poco imaginaba José Carlos Tomás cuando compró el Spectrum que aquel pequeño ordenador llegaría a convertirse en una herramienta más en su cotidiano trabajo. Gracias al programa VU-3D descubriría cómo potenciar el dibujo animado. Así nos lo decía desde su rincón de trabajo.

VU-3D

samos "S" y nos desplazamos hasta otro punto que esté unido con el anterior pulsando "L" a continuación, con lo que el programa nos unirá los dos con una recta. A continuación nos vamos a un tercero y volvemos a pulsar "L" para unirlo con el segundo y así sucesivamente. Cuando hayamos definido todos los puntos pulsamos "E" con lo que la figura se cierra uniendo el último punto definido con el primero y volvemos al menú de *"CREATE"* en el que podremos pasar a definir otra figura. Mientras estamos dibujando podemos pulsar la tecla "D" que nos borra la última figura definida (o en caso de que no hubiese ninguna definida nos volvería a *CREATE*. Una vez que hayamos definido todas las figuras de un plano z nos iremos desplazando por medio de la N hasta el plano donde se acaba cada figura (no es posible hacer que crezcan hasta el infinito) y aquí una vez seleccionada la figura con el comando *"Figure"* (que veremos a continuación) elegimos la opción *Close* que la cierra definiendo ese límite como el superior, al cerrar el programa nos preguntará si queremos que el borde superior sea abierto o cerrado (que deje pasar o no la luz). Para definir un dado que mida 20 puntos de cada lado deberemos, por tanto, colocarnos en el plano z = 0, abrir (con *Open*) la figura y dibujar la base usando las flechas y "S", "L", "D", una vez hecho esto damos una "E" para volver a *"CREATE"* y desde allí, pulsando veinte veces la N nos vamos al plano z = 20. Allí seleccionamos la opción Close y cerramos la figura, haciendo que tenga la misma forma en la parte superior que en la inferior. Si necesitamos crear varias figuras a la misma altura (las patas de una silla, por ejemplo) deberemos abrir una vez por cada una de ellas, dibujarla, salir usando "E" y abrir otra con la que se realiza el mismo proceso.

El comando *Figure* es útil cuando nos hemos abierto varios dibujos distintos y no los hemos cerrado después (cuando creamos las cuatro patas de la silla) pulsando la tecla "F" nos va cambiando la figura seleccionada, esa es la que cerramos al elegir el comando *Close* o la que podemos modificar con otros comandos, si sólo tenemos una figura abierta a la vez este comando no hace nada. La figura seleccionada se muestra dibujándola a trazos, mientras las demás lo son con líneas continuas.

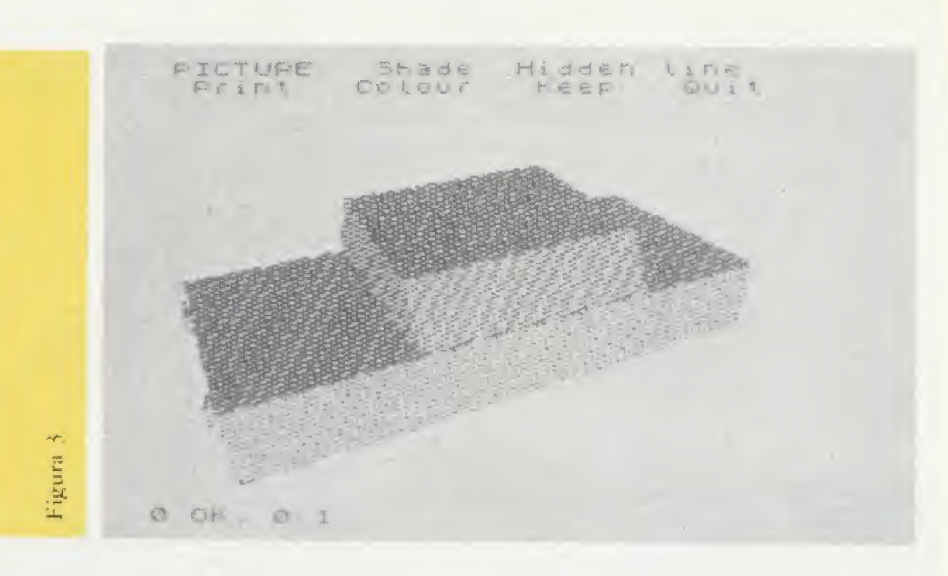


Figura 3

VU-3D en el cine

El dibujo animado es, sin duda alguna, una de las facetas que el cine utiliza para crear historias y personajes fantásticos que de otro modo sería impensable.

Básicamente la animación es muy simple. Sólo es cuestión de filmar con una cámara imagen por imagen una serie de dibujos entre los cuales hay ligeras diferencias. Si nos metemos de lleno en la técnica del dibujo animado veremos que es un poco más complicado pero el principio es el mismo. Todo depende del tipo de dibujo o de los objetos que queramos darle vida con nuestra cámara.

El año pasado tuve que hacer una película de presentación para unas proyecciones de cine amateur y se me ocurrió utilizar mi Spectrum para darle un toque de originalidad. Puesto que el VU-3D te permite mover figuras tridimensionales creadas por ti pensé rápidamente en utilizarlo para hacer parte de la película. Los resultados fueron cortos en cuanto a duración pero muy satisfactorios por la calidad obtenida.

Lo primero que hice fue dibujar en un papel, a grandes rasgos, la figura que quería que apareciese en pantalla. Era conveniente tener una idea previa antes de ponerme en el asunto. Una vez satisfecho con el resultado cargué el VU-3D y pulsé la opción "crear una figura". Aparecieron en la pantalla los ejes de coordenadas X e Y y los comandos que podía utilizar. Pulsé *OPEN* (Abrir) y el cursor hizo acto de presencia, al tiempo que cambiaban los comandos a utilizar. Puse el cursor en la parte inferior izquierda y comenzé a dibujar la figura como si la viera desde arriba. Más adelante le daría profundidad una vez centrada en la pantalla.

Tomé, no obstante, nota de todas y cada una de las coordenadas para poder repetirla en caso de salirme mal.

Los comandos *Magnify* y *Reduce* se usan junto con la figura seleccionada con *Figure* y nos aumentan o disminuyen el tamaño de la parte superior del dibujo en cuestión. Como la base permanece del tamaño que previamente le hemos definido, si reducimos el tamaño de la parte superior *(Reduce)* nos quedará una figura con forma de pirámide, en cambio si *magnificamos* nos quedará una pirámide invertida (similar a la parte superior de una copa). Al cerrar la figura, el programa une con líneas rectas las partes superior e inferior correspondientes. De modo que si queremos crear un lápiz deberemos hacer el cuerpo con un figura en la que la parte inferior y la superior tengan el mismo tamaño y, a continuación añadirle en la parte superior otra figura con la base del mismo tamaño pero la parte superior *Reducida* hasta hacer punta (Fig. 1). Además, desde el *CREATE* podemos mover la parte superior con las flechas, de modo que no se encuentre centrada respecto a la parte inferior, sino desplazada a un lado con lo que se pueden obtener efectos interesantes (el respaldo de nuestra silla, por ejemplo, se puede inclinar para que no esté recto, sino que la parte superior esté más para atrás que la inferior).

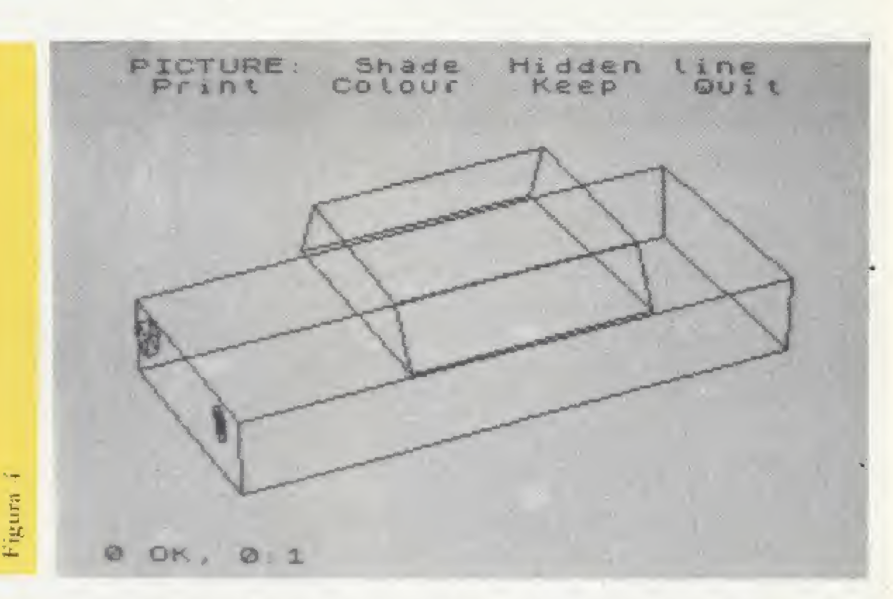


Figura 1

Una vez definido todo el objeto a representar debemos salir del menú *CREATE* haciendo uso de la opción *QUIT* (una Q) que nos conducirá al menú principal, en el que aparecen otras opciones además de la que acabamos de dejar (crear figura).

La primera opción de este menú es la de modificar una figura. Con ella podemos variar el tamaño y la posición en el plano de las distintas figuras, pero no su posición en el eje z, ni tampoco podemos quitar o añadir figuras. Podemos, por ejemplo, mover un dibujo de la parte inferior de la pantalla a la superior, pero no podemos hacer que empiece a una "altura" de 20 en vez de 10. El procedimiento para hacer esto es similar al utilizado durante la definición, por lo que no debe plantear ningún problema.

La opción de abandonar, no borra el programa como en un principio se puede pensar, sino que borra el dibujo existente en memoria y prepara el programa para la introducción de otro nuevo.

Otras dos opciones permiten leer y escribir en cinta un fichero de datos, esta opción no graba la pantalla para poder leerla posteriormente por este o por otro programa con una instrucción LOAD "" SCREEN$, sino que graba un conjunto de números que identifi-

Una vez terminada la fase creativa pulsé *QUIT* y volví al menú principal donde tecleé la opción *display*. Automáticamente salió en pantalla mi figura en tres dimensiones. Me acerqué a ella y la hice girar en todas las direcciones para asegurarme de su perfecto estado. Llegado este punto volví al menú y grabé en cinta los datos de la figura para una posterior utilización.

Al día siguiente me dispuse para la dura tarea de filmar. Previamente ya tenía una ligera idea de las evoluciones que mi diseño tenía que hacer en pantalla, por lo que fui directo al grano.

Coloqué la cámara frente al televisor y lo primero que hice fue ajustar el diafragma. Para ello era necesario encender el televisor y procurar que ninguna otra luz interfiriera. Quité la antena y aumenté el brillo y el contraste casi

"Con el VU-3D es fácil hacerse un taller de animación por ordenadores."

hasta el máximo. Medí la luz con un fotómetro y ajuste manualmente el diafragma, el cual no tocaría hasta terminar el trabajo. Hecho esto conecté el Spectrum y cargué el VU-3D. Después cargué los datos de la figura, almacenados el día anterior, y pulsé *display*. Eliminé de pantalla el cuadro de mandos de la parte de arriba poniendo el PAPER y la TINTA de igual color para que no me estorbaran. Aumenté el dibujo hasta que ocupó toda la pantalla y entonces centré y enfoqué la cámara. Esta tenía que quedar perfectamente a la altura del televisor y perpendicular al tubo. El enfoque lo hice poniendo el *zoom* en posición de tele y luego retrocediendo hasta el límite del recuadro central.

Una vez todo dispuesto y teniendo la precaución de cerrar la puerta de la habitación para que nadie me molestara, situé la figura en la

VU-3D

can los vértices de nuestra figura, de modo que podamos seguir trabajando posteriormente con ellos. Este fichero sólo puede ser leído por este programa.

La rutina denominada pantalla o *display* es la destinada a la representación, en tres dimensiones, del objeto. Una vez que entramos en ella, se borra la pantalla y aparece en la parte superior una lista (denominada *DISPLAY*) de todos los comandos disponibles a este nivel. Asimismo en la parte inferior aparece otra línea en la que se indican los siguientes datos: el grado de *magnificación* de la figura (si hacemos *zoom*), la rotación que le hemos dado a la figura (el ángulo desde el que la contemplamos) y el desplazamiento respecto al eje z. Aunque puede parecer que el *zoom* y el desplazamiento del eje z son similares, existen pequeñas diferencias a la hora de manejar la figura. Si acercamos demasiado el eje z, partes de la figura se deforman (puesto que se salen del, teórico, plano de la pantalla), en cambio moviendo el *zoom* conseguimos mayor acercamiento sin deformar la imagen, asimismo cambia (aunque sólo ligeramente) la perspectiva del objeto. Mientras el *zoom* no modifica las proporciones, sino que sólo amplía el tamaño. El desplazamiento del eje z, proporciona una mayor sensación de profundidad. En la figura 2 se ha acercado el objeto lo más posible acercando el eje z, mientras que en la figura 3 nos hemos alejado del objeto (aunque conservando el mismo punto de vista) y hemos utilizado el *zoom* para acercarlo. En esta última foto se nota una falta de perspectiva que hace que la imagen carezca de cuerpo.

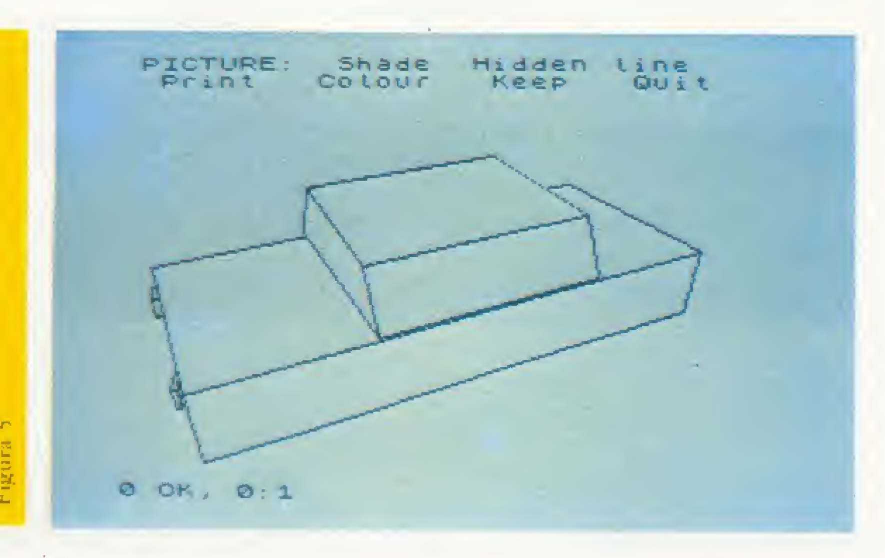


Figura 5

Los comandos disponibles a este nivel son: *Magnify*, que nos acerca el objeto haciendo uso del *zoom* comentado anteriormente. *Reduce* efectúa la maniobra contraria del *Magnify*, es decir, aleja el objeto pero no modifica la perspectiva. *Near* nos desplaza el objeto moviéndonos por el eje z. De este modo aumenta su tamaño y se crea una mejor sensación de perspectiva, pero, atención, si nos acercamos demasiado algunas partes del objeto pasarán del plano de la pantalla y el programa las cortará. El comando *Far* nos aleja del objeto deshaciendo el camino hecho con el *Near*. El comando *Quit* nos devuelve al menú principal y por último *Picture* nos pasa a un submenú donde nos encontramos otras siete opciones: *Quit* nos devuelve al menú anterior, *Shade* sombrea

VU-3D en el cine

fase inicial. Hice unas cuantas pulsaciones con mi disparador de cable para que la figura no se moviera al principio de la película. Después todo el proceso iba a ser repetitivo: filmaba otras dos imágenes y movía la figura una posición para volver a filmar otras dos imágenes. La línea inferior me daba cuenta del ángulo de rotación, así como de la proximidad de la figura, lo cual me era muy útil a la hora de hacer ensayos de movimiento durante la marcha, ya que anotaba las últimas posiciones para después poder regresar a ellas.

Otro truco que utilicé fue de el cambiar colores de papel y tinta para dar la impresión de un flash. El programa me permitía hacerlo con toda comodidad.

Después de toda una tarde de pulsar teclas y con los ojos en forma rectangular recogí cuidadosamente todos los trastos. Al día siguiente llevé la película al laboratorio. El trabajo había concluido.

Ya sea con el VU-3D o con otro tipo de programa de gráficos, he podido comprobar lo fácil que es hacerse un taller de animación por ordenador, en donde éste nos evita el tener que dibujar y utilizar tableros especiales y otros equipos auxiliares que esta técnica requiere. Sin llegar, ni mucho menos, a la perfección de los dibujos de ordenador profesionales, sí que podemos aventurarnos, conociendo siempre nuestros límites, en el maravilloso mundo de la animación.

Hace algunos años se estrenaba en las pantallas de nuestro país la película "TRON". Su historia era una de tantas, sin embargo la forma y concepción de la imagen era completamente distinta a otra idéntica en su género. La utilización de gráficos diseñados por computadora ocupaban gran parte de la historia. Eran por sí mismos protagonistas.

¿Se podrían hacer cosas parecidas con nuestro Spectrum?. Sí se

el objeto dependiendo de la posición de un foco teórico que puede estar colocado en nueve posiciones posibles dependiendo de su altura y si está hacia la izquierda, la derecha o en el centro, el dibujo adopta, a veces, resultados espectaculares, aunque la mayoría de las veces no salga tan bonito como se espera (Fig. 4). La opción *Hidden* borra todas las líneas que no se ven si las superficies se consideran opacas (y no de cristal como hasta ahora) pero no sombréa (Fig. 5). *Print* nos permite sacar una copia en papel impreso del dibujo (si disponemos de una impresora con capacidad gráfica). *Keep* permite salvar el gráfico en cinta de modo que pueda ser leído con la instrucción LOAD "" SCREEN$, ya que hace un volcado de la pantalla por impresora. Por último la opción *Colour* nos permite modificar los colores del borde, del papel del dibujo y de la zona de órdenes (independientemente) y de la tinta de estos dos (también independientemente), aunque en el programa que vimos no funcionaba correctamente (sólo cambia el color del *border*) y tuvimos que recurrir a una opción similar disponible en el menú principal que sí funcionaba correctamente. Uno de los usos más interesantes que se puede dar a la opción de los colores del panel de mandos es hacer que el color de la tinta y del papel sean similares al papel del dibujo, de modo que al salvarlo (o imprimirlo) no salgan las letras, sino el dibujo solamente (Fig. 6).

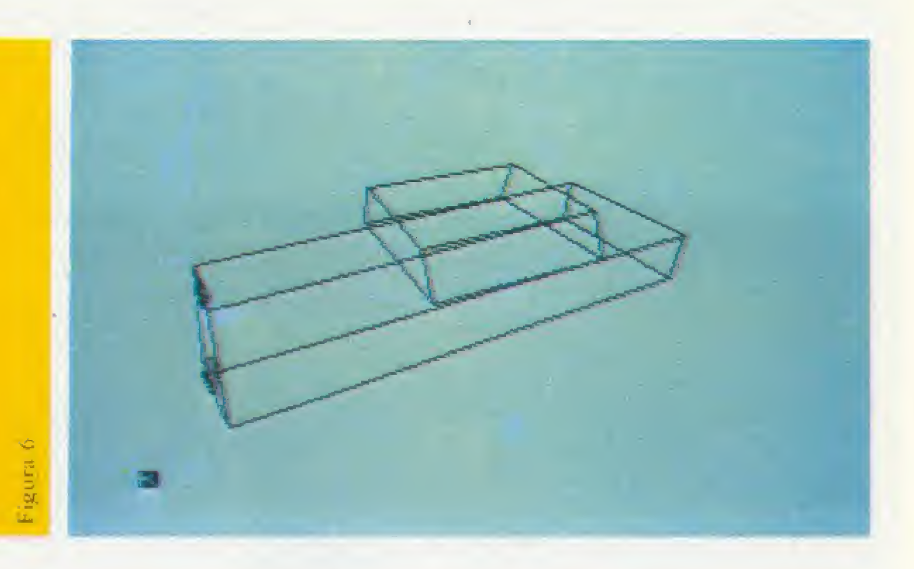


Figura 6

Cuando estamos en la opción *DISPLAY* (y siempre que no seleccionemos *PICTURE*) podemos girar la figura respecto al eje vertical y al horizontal de modo que podamos ver todas las partes del dibujo, para realizar esto se utilizan las flechas del cursor (sin necesidad de usar CAPS SHIFT, que en este programa se utiliza para aminorar el movimiento de las figuras).

El programa está bien desarrollado dando un resultado bueno con las representaciones aunque tarde demasiado en moverse, asimismo resulta un poco incómodo de manejo el sistema usado para introducir los dibujos ya que resulta poco claro de compresión y rígido de manejo (una ampliación de esta parte del programa podría consistir en la posibilidad de retroceder en el eje z y poder borrar partes del objeto que ya estén definidas, ya que actualmente sólo se pueden borrar cuando no se ha terminado de dibujar la parte inferior).

El programa resulta útil para aquellas personas que deseen realizar representaciones de objetos tridimensionales a partir de planos y esquemas.

---

puede, teniendo, claro está, tres cosas: una cámara cinematográfica (Super 8, por ejemplo), el Spectrum y un programa llamado VU-3D.

En rasgos generales dicho programa consiste en la creación de figuras tridimensionales de las cuales, gracias a una serie de comandos, se pueden girar en todas direcciones así como alejarnos o acercarlas a nosotros. También la figura creada por nosotros puede ser modificada lo que constituye otro efecto más a utilizar.

Personalmente mi *hobby* preferido, aparte del Spectrum, es el cine en su más amplio sentido. El año pasado me encargaron una película para la presentación de unas proyecciones de cine amateur. Me vino como caído del cielo el VU-3D para mis propósitos. Coloqué la cámara delante del televisor, conecté el Spectrum, cargué el programa y después de crear una figura me dediqué a filmar foto a foto cada evolución en pantalla.

"Entre desplazamientos es algo lento, pero las posibilidades son enormes."

Los resultados fuero buenos aunque su duración total no pasara de los 45 segundos a pesar de estar todas una tarde trabajando. Tengo que hacer notar que entre desplazamientos el programa resulta algo lento y más si se quiere rellenar o sombrear la figura. De todas formas las posibilidades dentro de este aspecto son bastantes y tan sólo limitadas por la creatividad de cada cual.

Lo que parecía un programa de "inutilidad" se convierte de esta forma en algo vivo e inédito y de largas horas de entretenimiento.

Llegó el

Antes de empezar a comentar la implementación del FORTH realizada en el Spectrum, conviene explicar los fundamentos de este lenguaje.

El FORTH es un lenguaje creado para ser rápido y que pueda ser implementado en máquinas pequeñas. Su estructura se halla a medio camino entre el código máquina y los lenguajes de alto nivel, básicamente sólo maneja números enteros como el código máquina, aunque llega hasta valores de 520090719, lo cual es realmente elevado, y también maneja tiras de caracteres y realiza otros procesos sólo existentes en lenguajes de alto nivel.

Su funcionamiento es una mezcla de compilador e intérprete y funciona basado en una pila (stack) de tipo LIFO, es decir, tiene una zona de memoria donde se almacenan los datos secuencialmente, siendo el último dato escrito el primero en ser usado. Esta zona de memoria la amplía según lo necesite. El FORTH no se programa como entendemos habitualmente, lo que se hace es definir nuevas "palabras" y "diccionarios" que permiten ejecutar las funciones que uno desea.

La versión de FORTH del Spectrum está diseñada para el aprendizaje ya que tiene dos características no standard. La primera es que está continuamente comprobando si se le pulsa el BREAK; esta característica, aunque normal en el BASIC, no es normal en el FORTH; que es un lenguaje pensado para la rapidez y esta comprobación le hace perder tiempo, aunque al usuario novato le permita equivocarse sin obtener resultados catastróficos. La segunda característica es la existencia de una serie de palabras que permiten comprobar el estado del sistema desde dentro de una "palabra" que se esté ejecutando.

Para cargar el programa se sigue el procedimiento habitual de LOAD" "CODE y una vez cargado se ejecuta automáticamente. En la cinta viene, además, un editor de FORTH escrito en



FORTH, que se carga desde dentro del programa usando las palabras 1 LOAD. (debemos indicar que en el manual hay muchas palabras que vienen unidas; en este caso aparece 1LOAD siendo incorrecto ya que el intérprete no lo comprende y dará error).

Una vez puestos a trabajar en FORTH y con el editor cargado ya estamos preparados para escribir nuestra primera palabra:

Tecleamos:

: PRUEBA 999 0 DO I . LOOP ;

Si ahora pulsamos ENTER y tecleamos PRUEBA (ENTER) veremos imprimirse en la pantalla los números del 0 al 998 uno a continuación de otro y separados por un espacio. Cada vez que usemos la palabra PRUEBA en una definición o bien usada directamente se compilará o ejecutará respectivamente el bloque que acabamos de definir. Estos bloques no tienen porqué ocupar una sola línea, es más, este lenguaje tiene estructura libre y por tanto podríamos haber escrito una palabra en cada línea. Si no quisiéramos usar esta "palabra" nunca más y queremos que nos deje libre su espacio de memoria, bastará decir "FORGET PRUEBA" seguido de ENTER, lo que le hará olvidar, no sólo la palabra prueba sino todas aquellas palabras definidas posteriormente en el mismo diccionario, así que tenga cuidado con lo que olvida.

Un diccionario es una lista de

FORTH



palabras donde el FORTH busca el significado de las órdenes que se le dan. El sistema nos permite tener varios diccionarios distintos a la vez. La razón de que se puedan usar varios o no uno solo donde englobemos a todas las palabras es que así disponemos de la posibilidad de tener bloques independientes (editor, ensamblador, números reales, etc) que pueden ser olvidados cuando no los necesitemos y queramos aprovechar su espacio. También nos permite disponer de palabras distintas pero con el mismo nombre (una en cada diccionario); por ejemplo el diccionario FORTH tiene la palabra "I" que lo que hace es poner en la pila la variable de un lazo, mientras que el diccionario EDITOR tiene la palabra "I" que lo que hace es insertar textos. Para usar un diccionario u otro debemos escribir el nombre del diccionario que queremos usar seguido de la tecla ENTER, el sistema nos hará abandonar el diccionario en que estemos ahora y entrar en el otro, a partir de ese momento sólo nos recomendará las palabras del nuevo diccionario.

Para saber las palabras que tenemos en el diccionario en el que nos hallamos, nos basta con teclear "VLIST" lo que hará que aparezca una lista de todas las palabras en orden inverso al de introducción. Este listado puede ser parado pulsando BREAK.

Para definir una palabra de la que sepamos que es correcta empleamos el siguiente sistema: primero debemos teclear ":" esto indica al intérprete que se va a definir una palabra. A continuación se escribirá el nombre de la palabra (máximo de 31 caracteres) seguido de un espacio. Las siguientes palabras que se escriban ya será la secuencia de órdenes a ejecutar cuando sea llamada esa palabra. Al acabar la definición debemos poner un punto y coma para indicar que la definición está completa y devolver el control al intérprete. Recordamos aquí que las palabras FORTH deben estar separadas por espacios para su correcta interpretación por el sistema.

Este lenguaje trabaja con cinco tipos de datos:

- CARACTERES Y BYTES: Ocupan 8 bits (1 byte).
- NUMEROS CON SIGNO: Ocupan 16 bits y sus valores se hallan comprendidos entre -32767 y +32767.
- NUMEROS CON SIGNO: Ocupan también 16 bits, pero varían entre 0 y 65535.
- NUMEROS DOBLES CON SIGNO: Ocupan 32 bits y están comprendidos entre -2147483647 y +2147483647.
- NUMEROS DOBLES SIN SIGNO: Así mismo ocupan 32 bits y varían entre 0 y 4294967296.

En FORTH no hace falta definir explícitamente el tipo de dato que se va a usar, ya que la palabra impone el tipo de operando a usar, llegando incluso a usar cada una de las mitades de un número doble como número simple si así se lo pedimos.

Las palabras básicas del FORTH se pueden dividir en bloques: Las de manejo de stack, aritméticas, comparación, manejo de memoria y las de control, aparte de otras que no están definidas en ningún grupo.

Las palabras de manejo de stack están pensadas para alterar el orden lógico del stack o duplicar los datos. La forma en que un número se mete en el stack depende de su tipo, para meter un número simple (de uno o dos bytes, con 0 sin signo) basta con escribirlo entre dos espacios, los dobles (de 32 bits, con o sin signo) son introducidos con el mismo procedimiento, pero añadiéndole un punto en cualquier parte del número, hay que aclarar que este punto no tienen ningún significado matemático y que el FORTH considera el número entero como si no hubiésemos puesto el punto. Los números introducidos pueden ser manejados mediante "palabras" previstas a tal efecto, las más importantes son: *(ver figura 1)*

DUP: copia el número que se halla en la parte inferior de la pila colocándolo detrás de sí mismo.

ROT: gira los tres últimos números del stack.

-DUP: ejecuta la misma acción que DUP, excepto cuando el número sea cero en cuyo caso no hace nada.

.: el punto imprime el último número del stack y lo borra de éste.

SWAP: intercambia entre sí los dos últimos números del stack.

OVER: copia el penúltimo número en el final del stack.

Llegó el FORTH

Los operadores aritméticos, como su nombre indica, realizan operaciones aritméticas con uno o dos números, borrando los operadores y escribiendo el resultado en su lugar. *(Ver figura 2).*

+: suma los dos últimos números en el stack.

–: resta el último número al penúltimo.

*: multiplica los dos últimos números en el stack (¿qué se pensaba?).

/: realiza la división del penúltimo número partido por el último.

MOD: deja el resto de la división.

/MOD: deja el resto y el cociente de la división.

ABS: deja el valor absoluto (positivo) del último número del stack.

AND: realiza un Y lógico entre los bits de los dos últimos números del stack. Es decir, coge parejas de bits (uno de cada número) y si ambos son 1 el resultado es 1, en cualquier otro caso el resultado es cero.

OR: realiza un proceso similar al AND, pero en vez de hacer un Y lógico, realiza un 0: si ambos bits son 0, el resultado es cero, en cualquier otro caso el resultado es 1.

XOR: similar a las dos anteriores, pero el resultado es 1, sólo si los dos bits son distintos (uno vale 1 y el otro 0).

Para comprender mejor estas tres últimas operaciones se han dibujado en la *figura 3* las tablas de verdad de estas operaciones, donde se pone el valor de los bits de entrada y el de salida.

Los operadores de comparación actúan sobre dos números realizando un proceso de evaluación matemática, es decir, miran si es menor, mayor, igual, etc., desde un punto de vista matemático y no lógico como son las tres operaciones anteriores.

< : compara el último número con el penúltimo, borrándolos del stack y dejando 0 si el último no es menor y cualquier otro número en caso contrario.

> : su funcionamiento es similar al anterior, pero ahora comprueba si el último es mayor que el penúltimo.

=: comprueba, de un modo similar a las dos palabras anteriores, si los dos últimos números del stack son iguales.

0< : compara el último número del stack con 0, dejando 1 si, efectivamente es menor que 0, y cualquier otro número en caso contrario. Su funcionamiento es equivalente a escribir "0 < ".

0=: deja 0 si el último número es distinto de 0, y viceversa. Es equivalente a escribir "0 =".

Hay que hacer notar que en FORTH, al 0 nacido de una comparación se le puede llamar "bandera falsa" o, abreviadamente, "ff", y cualquier 0 puede funcionar como "ff". Los números distintos de cero son llamados "bandera verdadera" ("tf"). Más adelante veremos la utilidad de estas banderas.

Para manejar la memoria (usando como intermediario el stack, como va siendo habitual) se usan las siguientes palabras:

@: coge el último número del stack, lo interpreta como una dirección, y mete en el stack el número de 16 bits contenido en dicha dirección.

!: toma el penúltimo número del stack y lo almacena en el lugar de memoria apuntado por el último.

Figura 1. Palabras de manejo del stack.

DUP

Antes	Después
27	27
534	27
1212	534
	1212

DUP

a)

Antes	Después
23	23
15	23
	15

b)

Antes	Después
0	0
1234	1234

ROT

Antes	Después
1534	5223
5223	12
12	1534
515	515

OVER

Antes	Después
1367	–534
–534	1367
2222	–534
	2222

SWAP

Antes	Después
294	–150
–150	294
33	33

Figura 2. Operadores aritméticos.

+(SUMA)

Antes	Después
15121	18648
3527	12
12	

–(RESTA)

Antes	Después
15121	–11594
3527	12
12	

* (PRODUCTO)

Antes	Después
323	2261
7	391
391	

/ (DIVISION)

Antes	Después
254	15
16	49
49	

MOD

Antes	Después
255	14
16	49
49	

/MOD

Antes	Después
255	15
16	14
49	49

ABS

a)

Antes	Después
25	25
13	13

b)

Antes	Después
–25	25
13	13

C@: Su función es similar a la que realiza la @, pero con la variación de que el número leído es de 8 bits y no 16.

C!: almacena en la dirección de memoria indicada por el último número el byte inferior del penúltimo número del stack.

+!: esta curiosa instrucción coge el número almacenado en la memoria indicada por el último número del stack, le suma el penúltimo y almacena el resultado en la misma dirección. Hay que indicar que las operaciones intermedias no modifican el stack, aunque, naturalmente los dos números usados para la operación son borrados.

El último grupo de instrucciones del FORTH lo forman las de control, estas "palabras" forman, quizás, la parte más importante del FORTH ya que nos permiten tomar decisiones y obrar en consecuencia al resultado. Su uso se restringe a la definición de palabras, no pudiendo usarse directamente ya que dan error, por ejemplo:

: CUENTAMIL 1001 1 DO LOOP:

es legal, ya que estamos definiendo una palabra que será llamada posteriormente. En cambio:

1001 1 DO LOOP

daría el error: "DO? ERR MSG#17" ya que le estamos pidiendo que la ejecute de modo directo. En este punto cabe destacar que el BASIC del Spectrum sí nos permite ejecutar una instrucción similar:

FOR I = 1 TO 1000 : NEXT I

sin dar error.

Las estructuras de control del FORTH son las siguientes:

DO LOOP: se escribe con el formato:

(valor final +1) (valor inicial) DO (secuencia a ejecutar) LOOP

Realiza la secuencia tantas veces como indique la diferencia entre el valor final y el valor inicial, cada vez que pasa por el bucle incrementa el valor inicial en una unidad. La secuencia se ejecuta una vez al menos (incluso en el caso de que el valor final sea menor que el inicial).

DO +LOOP: es de funcionamiento similar al anterior con la diferencia de que se le indica lo que se debe incrementar el valor inicial. Su sintaxis es del tipo:

(valor final + incremento) (valor inicial) DO (secuencia a ejecutar) (incremento) +LOOP

IF ELSE THEN: es una estructura de bifurcación en la cual se realiza una acción u otra en función de una condición. Su estructura es:

(condición) IF (secuencia a ejecutar si la condición es cierta "tf") ELSE (secuencia si la condición es falsa "ff") THEN

La condición es un número (que puede ser el resultado de una operación) que consideramos falso "ff" si vale 0, y cierto "tf" en cualquier otro caso.

BEGIN UNTIL: esta estructura ejecuta repetidamente una secuencia de órdenes hasta que se cumpla una condición. Se escribe con la siguiente sintaxis:

BEGIN (secuencia a ejecutar) (condición) UNTIL

En este caso, al igual que los anteriores y posteriores, la condición es un número del tipo "ff" o "tf", además se ejecuta una vez aunque la condición fuese cierta desde un principio.

BEGIN WHILE REPEAT:: esta estructura se escribe:

BEGIN (secuencial) (condición) WHILE (secuencia2) REPEAT

su funcionamiento es similar al de la anterior con la diferencia de que la condición de control se encuentra en cualquier punto de la secuencia y no al final, saliendo del bucle por ese punto, por tanto si la ponemos antes de la secuencia, el FORTH mirará primero la condición y en función de eso ejecutará, o no, las instrucciones. Otra diferencia se encuentra en que para salir del bucle se tiene que cumplir que la condición sea "ff" (al contrario que en el caso anterior en el que la condición de salida es "tf").

BEGIN AGAIN: este bloque (también conocida como "piérdete") ejecuta la secuencia infinitas veces (o hasta que lo paremos por la fuerza bruta). Su sintaxis es:

BEGIN (secuencia) AGAIN

LEAVE: esta palabra no lleva estructuras anejas, cuando se ejecuta, da a la variable del lazo el valor adecuado para que el bucle termine con el siguiente paso por una instrucción "LOOP" o "+LOOP". En caso de que existiesen varias estructuras anidadas, sólo saldría del lazo más interno *(ver figura 4)*.

QUIT: esta es una palabra de

Figura 3. Operaciones lógicas.

Operación AND

Bits de entrada		Bit de Salida
Ø	Ø	Ø
Ø	1	Ø
1	Ø	Ø
1	1	1

	Representación decimal	Representación binaria
	55 =	Ø Ø 1 1 Ø 1 1 1
	23 =	Ø Ø Ø 1 Ø 1 1 1
55 AND 23 =	23 =	Ø Ø Ø 1 Ø 1 1 1

Operación OR

Bits de entrada		Bit de Salida
Ø	Ø	Ø
Ø	1	1
1	Ø	1
1	1	1

	Representación decimal	Representación binaria
	217 =	1 1 Ø 1 1 Ø 1 1
	31 =	Ø Ø Ø 1 1 1 1 1
217 OR 31 =	255 =	1 1 Ø 1 1 1 1 1

Operación XOR

Bits de entrada		Bit de Salida
Ø	Ø	Ø
Ø	1	1
1	Ø	1
1	1	Ø

	Representación decimal	Representación binaria
	224 =	1 1 Ø 1 1 1 1 Ø
	193 =	1 1 Ø Ø Ø Ø Ø 1
224 XOR 193 =	63 =	Ø Ø Ø 1 1 1 1 1

Llegó el FORTH

tipo similar a la anterior. Lo que hace es abandonar todas las estructuras y volver al intérprete

Si usted quiere empezar a trabajar en FORTH, le conviene definir la siguiente palabra:

```
: .S
SP @ SO @ -
-DUP 0=
IF
CR. "STACK VACIO"
ELSE
SO @
DUP ROT +
DO
  CR I ?
  2
+LOOP
THEN ;
```

Su utilidad (fundamental para principiantes) nos enseña el contenido del stack sin modificarlo y en caso de que estuviese vacío imprime "STACK VACIO". Para aquellas personas acostumbradas al BASIC u otros lenguajes similares les enseña como se modifica la pila en función de las operaciones que realicemos.

Incluimos a continuación unos listados en FORTH y en BASIC indicando su tiempo de ejecución, para poder comprobar las diferencias entre ambos lenguajes.

1) Contar de 1 a 10.000:

BASIC:

10 FOR I= 1 TO 10000 : NEXT I (segundos)

FORTH:

: TEST 10001 1 DO LOOP ; (segundos)

2) Calcula los cuadrados de los primeros 1000 números imprimiéndolos.

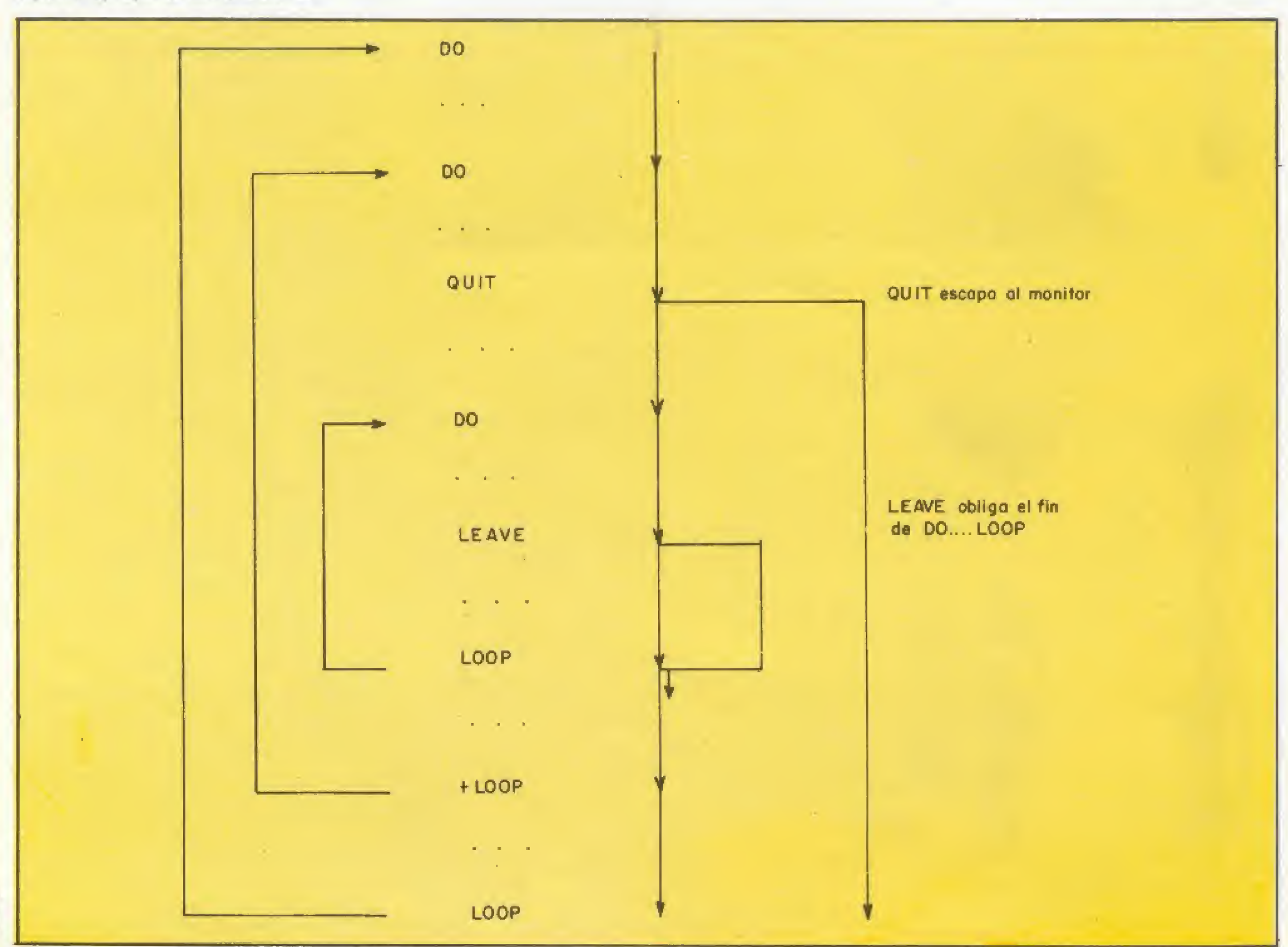
BASIC:

10 FOR I = 1 TO 1000 : POKE 23692,255 : REM Para que no pida scroll?
20 PRINT I^2: NEXT I (segundos)

FORTH:

: TEST 1001 1 DO I DUP M* D. LOOP ; (segundos)

Figura 4. Ejemplo del uso de LEAVE.



DINAMIC

SAIMAZOOM

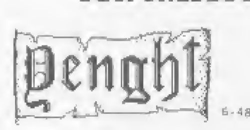
MAPSNATCH

VR SNATCHO

JOHNY JONES HERMANO DEL FAMOSO INDIANA, ES EL HEROE DEL SAIMAZOOM.
LA COMPAÑIA A LA QUE PERTENECE SOLO PODIA CONFIAR EN EL PARA ESA MISION:
SELECCIONAR EL MEJOR CAFE A LO LARGO Y ANCHO DE LA SELVA.
JOHNY DEBERA AFRONTAR SOLO LOS INNUMERABLES PELIGROS QUE LE ESPERAN:
NEGROS CANIBALES, SERPIENTES VENENOSAS, TEMPLOS EN RUINAS, VEGETACION EXUBERANTES, ETC.
LAS UNICAS ARMAS DE JOHNY SON SU PERICIA, VALOR Y BUENO, POR QUE NO DECIRLO, DE UN REVOLVER, PICOS, CANOAS, ETC.
SI JOHNY LO LOGRA SERA RECOMPENSADO CON...
...PERO MEJOR SERA QUE LO DESCUBRAS TU MISMO.
"LOS SPECTRUMANIACOS DISFRUTARAN COMO LOCOS" ZX OCTUBRE 1984.

AÑO 2000. EUROPA HA ENTRADO EN GUERRA.
LOS ARSENALES TERMONUCLEARES ESTACIONADOS EN EL CONTINENTE COMIENZAN A SER UTILIZADOS.
MORTIFERAS RAMPAS DE LANZAMIENTO DE MISILES SE ABREN PARA EXPULSAR SUS CARGAS LETALES.
TU PUEDES SER PARTICIPE Y TESTIGO DEL HOLOCAUSTO NUCLEAR.
EN ESTE WARGAME EXISTE UN HUECO PARA LA ESPERANZA:
SI CONSIGUES ORGANIZAR TUS ATAQUES Y DISTRIBUIR LOS EJERCITOS PODRAS ALCANZAR LA VICTORIA.
CON MAPSNATCH CONSEGUIRAS QUE TODO ESTO OCURRA SOLO DENTRO DE TU SPECTRUM.
"EL JUEGO RESULTA MUY DIVERTIDO Y DE GRAN INTERES" ZX OCTUBRE 1984

DISPONIBLES:



Denght 5-48 K

ARTIST 5-48 K

SAIMAZOOM 5-48 K

MAPSNATCH 5-48 K

MUY PRONTO:

PRO-COMPETITION 5-48 K

BABALIBA 5-48 K

VIDEO OLIMPIC 5-48 K

PROFANATION 5-48 K



DINAMIC

MONTEPRINCIPE, A-4. 21. BOADILLA DEL MONTE. MADRID.

BOLETIN DE PEDIDO.
DESEARIA RECIBIR:

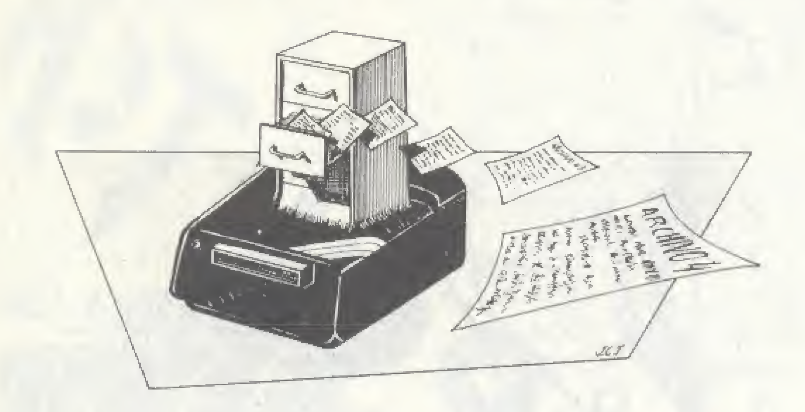
☐ YENGHT SAIMAZOOM ☐
☐ ARTIST MAPSNATCH ☐
INFORMACIÓN ☐

DINAMIC PONE A TU DISPOSICIÓN UN NUEVO DEPARTAMENTO DE VENTA POR CORREO QUE INCLUYE LAS SIGUIENTES MEJORAS:

1º PAQUETES NORMALIZADOS DE ALTA CALIDAD.
2º SERVICIO DE ATENCIÓN Y ENVÍO DE PEDIDOS INSTANTÁNEOS. DOS DÍAS A PARTIR DE LA RECEPCIÓN DE TU CARTA DE PEDIDO.
3º PRECIO IGUAL QUE EN TIENDAS: 1800 PESETAS, INCLUYENDO GASTOS DE ENVÍO.
4º PAGO CONTRAREEMBOLSO.

6 DINAMICS

Base de datos en microdrive



Lograr un buen manejo de datos a través de memorias externas, en nuestro caso *microdrives*, es el gran sueño de cuantos hemos de luchar contra ingentes y desorganizadas cantidades de datos. Con este programa podrá grabar sus datos en *microdrive* o *cassette*, según la opción deseada.

Le proponemos un simple programa de base de datos para el *microdrive* principalmente, que le permitirá almacenar e intercalar dos listas de ficheros. Unos ejemplos posibles serían el catalogar artículos de revistas, listas de *software* y números de teléfonos.

Teclee el programa tal y como aparece en el listado, y pulse RUN 500 y ENTER. Aparecerán las distintas opciones posibles. Veámoslas en forma resumida:

1. Añadir al fichero

En esta opción habrá de introducir el nombre y los datos. A continuación se le pregunta si están correctos. Pulsando "S" podrá añadir más datos, "N" borra la última entrada y permite teclear la entrada correcta y "F" finaliza la entrada y las intercalaciones de datos volviendo al menú principal.

```
   1 REM ***********************
   2 REM *     Programa de     *
   3 REM *    Base de datos    *
   5 REM ***********************
```

```
   9 REM rutina SORT
  10 PRINT AT 20,0; FLASH 1;"ESP
ERE POR FAVOR, ESTOY ORDENAN-DO.
"
  20 LET t=0
  30 FOR i=1 TO c-1
  40 IF a$(1,i)<=a$(1,i+1) THEN
GO TO 70
  49 REM intercambiar
  50 FOR j=1 TO 2: LET s$=a$(j,i
): LET a$(j,i)=a$(j,i+1): LET a$
(j,i+1)=s$: NEXT j
  60 LET t=1
  70 NEXT i
  80 IF t=1 THEN GO TO 20
  90 BEEP 1,0: RETURN
```

Rutina de Sort. "c" es la longitud del archivo y "t" es una variable de control para detectar el final del Sort.

```
  99 REM Rutina para catalogar e
n z$ el cartucho, d=num. drive
 100 CLOSE #4: OPEN #4;"m";d;"in
dex.d"
 110 CAT #4,d
 120 CLOSE #4
 130 LET z$="": LET eof=0
 140 OPEN #4;"m";d;"index.d"
 150 INPUT #4;r$
 160 IF r$="" THEN LET eof=eof+1
: IF eof=2 THEN GO TO 150
 170 LET z$=z$+r$
 180 IF eof<>2 THEN GO TO 150
 190 CLOSE #4
 200 ERASE "m";d;"index.d"
 210 RETURN
```

Rutina para introducir el catálogo en la variable Z$.

```
 499 REM Procedimiento inicial
 500 DIM a$(2,1000,15): DIM d$(1
5): LET pr=2: LET c=1
```

Inicialización tablas y variables de control.

2. Borrar el fichero

Teclee el nombre del fichero a borrar. El nombre y los datos de este fichero quedarán anulados.

3. Búsqueda y listado de elementos

Teclee el nombre de la cadena a buscar dentro de la lista y todos los elementos que contengan la cadena se listarán. Esta es una de las partes más importantes del programa, consecuencia del rápido algoritmo de búsqueda que se describe en el listado. Observe el bucle doble que aparece en las líneas 3060 a 3100.

```
 510 PAPER 1: BORDER 1: CLS : IN
K 7: PRINT AT 5,10;"© G.B.MONK":
 PAUSE 100: CLS
 519 REM opciones principales
 520 CLS : PRINT AT 2,2;"ELIGE U
NA OPCION"
 530 PRINT AT 6,2;"1-ANYADIR AL
FICHERO"
 540 PRINT AT 8,2;"2-DELETE DEL
FICHERO"
 550 PRINT AT 10,2;"3-BUSQUEDA/L
ISTAR ELEMENTOS"
 560 PRINT AT 12,2;"4-IMPRIMIR F
ICHERO"
 570 PRINT AT 14,2;"5-IMPRESORA
";"ON" AND pr=3;"OFF" AND pr=2
 580 PRINT AT 16,2;"6-SAVE/LOAD
DATOS"
 590 PRINT AT 20,6; FLASH 1;"Pul
sa el NUMERO deseado."
 600 LET k$=INKEY$
 610 IF k$<"1" OR k$>"6" THEN GO
 TO 600
 620 BEEP .3,15: CLS : GO SUB 10
00*VAL k$: GO TO 520
```

Menú. Opciones del programa.

```
 999 REM anyadir al fichero
1000 IF c=1000 THEN PAPER 0: INK
 7: BORDER 0: CLS : PRINT TAB 8;
 FLASH 1; BRIGHT 1;"LO SIENTO, N
O HAY SITIO.": PAUSE 1: PAUSE 20
0: BORDER 1: PAPER 1: INK 7: RET
URN
1010 PRINT AT 20,0;TAB 5;"NOMBRE
";TAB 20;"DATOS"'': RANDOMIZE US
R 3582
```

Rutina de altas del fichero. La línea 1000 comprueba si el fichero se encuentra lleno. La instrucción RANDOMIZE USR 3582 produce el SCROLL de una línea. La línea 1070 permite la introducción e impresión de nuevos datos.

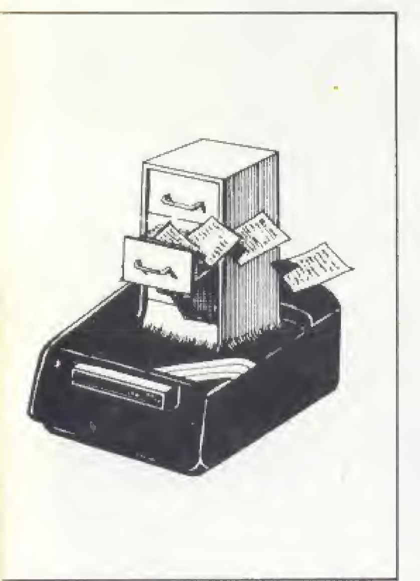
Base de datos en microdrive

4. Imprimir ficheros

Se imprime el fichero completo, bien por pantalla o bien por impresora dependiendo de la opción que se haya seleccionado en el punto siguiente.

5. Impresora ON/OFF

Permite la obtención de informes por pantalla o por impresora. Inicialmente se realiza la im-



presión por pantalla. Presionando la opción 5, aparecerá un mensaje en la parte superior izquierda de la pantalla "IMPRESORA ON" o "IMPRESORA OFF" según el caso.

```
1020 LET c=c+1: REM incrementa e
l contador del fichero
1030 INPUT "Nombre? ";a$(1,c)
1040 PRINT AT 20,0;a$(1,c),
1050 INPUT "Datos? ";a$(2,c)
1060 PRINT AT 20,15,a$(2,c)
1070 PRINT AT 21,0;"O.K. (s/n) o
r p to STOP."
1080 IF INKEY$="n" OR INKEY$="N"
 THEN GO TO 1030
1090 IF INKEY$="s" OR INKEY$="S"
 THEN PRINT AT 21,0;"
                  ": RANDOMIZ
E USR 3582: GO TO 1020
1100 IF INKEY$<>"p" AND INKEY$<>
"P" THEN GO TO 1080
1110 RANDOMIZE USR 3582: GO SUB
10: REM sort
1120 RETURN
```

```
1999 REM delete del fichero
2000 LET d=0: REM d=delete conta
dor
2010 INPUT "Elemento a borrar? "
;d$
2020 FOR i=1 TO c
2030 IF a$(1,i)=d$ THEN LET a$(1
,i)=CHR$ 143: LET d=d+1
2040 NEXT i

2050 GO SUB 10: REM sort
2060 LET c=c-d
2070 RETURN
```

La línea 2030 produce un espacio en INVERSE (CHR$ 143), a fin de que quede colocado en último lugar la realización del Sort. Por efecto de las líneas 2050 y 2060, se llama a la rutina de Sort y se decrementa el contador de archivos en una unidad, con lo que en definitiva se elimina el último fichero.

```
2999 REM buscar e imprimir
3000 LET t=1
3010 INPUT "Elemento a buscar ?
 ";s$
3020 PRINT AT 19,0;"Buscando a "
;s$
3030 PRINT #pr;AT 20,0;TAB 2;"EL
EMENTO";TAB 20;"DATOS"
3040 IF pr=3 THEN LPRINT
3050 LET ln=LEN s$-1
3060 FOR i=1 TO c
3070 FOR j=1 TO 15-ln
3080 IF a$(1,i,j TO j+ln)=s$ THE
N RANDOMIZE USR 3582: LET t=0: P
RINT #pr;AT 21,0;a$(1,i),a$(2,i)
3090 NEXT j
3100 NEXT i
3110 RANDOMIZE USR 3582
3120 IF t=1 THEN PRINT AT 21,0;"
No existe tal elemento.": RANDOM
IZE USR 3582
3130 IF pr=2 OR t=1 THEN PRINT A
T 21,0;"Pulse s cuando este list
o.": IF INKEY$<>"s" THEN GO TO 3
130
3140 RETURN
```

La variable "t" permite saber si se ha encontrado el fichero. Las líneas 3050 a 3100 constituyen la rutina de búsqueda.

```
3999 REM imprimir fichero
4000 PRINT #pr;AT 20,0;TAB 4;"EL
EMENTO";TAB 20;"DATOS"
4010 IF pr=3 THEN LPRINT
4020 FOR i=1 TO c
4030 RANDOMIZE USR 3582
4040 PRINT #pr;AT 21,0;a$(1,i),a
$(2,i)
4050 NEXT i
4060 PAUSE 1: PAUSE 100
4070 RETURN
```

Impresión del fichero.

```
4999 REM impresora on/off
5000 IF pr=2 THEN LET pr=3: PRIN
T "IMPRESORA ON": PAUSE 1: PAUSE
 50: RETURN
5010 IF pr=3 THEN LET pr=2: PRIN
```

Posibilidad utilización impresora.

6. SAVE/LOAD

Con esta opción se permite el almacenamiento del programa y los datos en *cassette* o *microdrive*. Si se elige la opción del *microdrive* se comprobará si ya existe en el cartucho, así como la existencia de espacio suficiente para la grabación. La rutina a catalogar en Z$ hace estos pasos y comprueba los posibles errores.

Con el formato actual, el programa se puede utilizar para guardar tipos múltiples de datos.

```
ELIGE UNA OPCION

1-ANYADIR AL FICHERO
2-DELETE DEL FICHERO
3-BUSQUEDA/LISTAR ELEMENTOS
4-IMPRIMIR FICHERO
5-IMPRESORA OFF
6-SAVE/LOAD DATOS

    Pulsa el NUMERO deseado.
```

Menú de opciones (líneas 519 a 580)

```
  ELEMENTO          DATOS

. Ruiz            cibernetica
uan Ar.           arcade games
.John             machine code
amirez            circuitos
odolfo            digitos
antiago           sprites
```

Ejemplo de impresión por pantalla o impresora, según opción elegida.

Sin embargo, el programa es muy manejable e incluye una potente rutina de búsqueda. Esta rutina (líneas 99 a 210) puede resultar muy útil para incorporar en sus programas que utilicen *microdrive*.

```
T "IMPRESORA OFF": PAUSE 1: PAUS
E 50: RETURN
```

```
5999 REM save/load
6000 PRINT AT 2,4;"ELIJA OPCION"
''
6010 PRINT TAB 4;"1-CASSETTE"''
6020 PRINT TAB 4;"2-MICRODRIVE"
6030 LET k$=INKEY$
6040 IF k$<>"1" AND k$<>"2" THEN
 GO TO 6030
6050 IF k$="2" THEN GO TO 6500
```

Opción cassete o microdrive.

```
6059 REM cassette
6060 CLS
6070 PRINT "Prepare el CASSETTE
de datos."
6080 PRINT "Desconecte EAR."
6090 SAVE "Data Base" LINE 510
6100 CLS
6110 PRINT "Verificacion. Rebobi
ne la cinta y conecte EAR."
6120 PRINT "Si hay algun error "
" GOTO 6060 """
6130 PRINT "Ponga en marcha el c
assette."
6140 VERIFY "Data Base"
6150 PRINT "O.K."
6160 PAUSE 1: PAUSE 200
6170 RETURN
```

Rutina grabación en cassette.

```
6499 REM microdrive
6500 CLS : PRINT " Numero del dr
ive ? (1-8)": FOR i=1 TO 50: NEX
T i
6510 LET k$=INKEY$
6520 IF k$<"1" OR k$>"8" THEN GO
 TO 6510
6530 LET d=VAL k$
6540 CLS : PRINT " Numero del dr
ive ";d
6550 INPUT " Nombre del fichero
?";s$
6560 IF LEN s$>10 OR LEN s$=0 TH
EN GO TO 6550
6570 PRINT "Nombre del fichero:
";s$
6580 PRINT ''"Leyendo el catalog
o. Espere"
6590 GO SUB 100
6600 LET z$=z$(11 TO )
6610 LET s=0
6630 IF z$( TO LEN s$)=s$ THEN L
ET s=1: GO TO 6660
6640 LET z$=z$(11 TO )
6650 IF LEN z$>10 THEN GO TO 663
0
6660 IF s=1 THEN GO TO 6800
6670 IF VAL z$<35 THEN PRINT "No
 hay espacio suficiente.": PAUSE
 1: PAUSE 200: RETURN
6680 PRINT "Grabando ";s$
6690 SAVE *"m";d;s$ LINE 510
6700 PRINT s$;" Grabado."
6710 PAUSE 1: PAUSE 100
6720 RETURN
6800 PRINT #1;AT 0,0;"El fichero
 ya existe -Lo susti- ra(s/n)"
6810 LET k$=INKEY$
6820 IF k$="N" OR k$="n" THEN RE
TURN
6830 IF k$<>"S" OR k$<>"s" THEN
GO TO 6810
6840 PRINT "Borrando Fichero ";s
$
6850 ERASE "m";1;s$
6860 GO TO 6680
9999 CLEAR : ERASE "m";1;"Dbase
2.3": SAVE *"m";1;"Dbase 2.3"
```

Rutina de grabación en microdrive.
Líneas 6500 a 6570 permiten la identificación del drive y el nombre del archivo.
Línea 6590: llamamiento a la rutina de impresión del catálogo en Z$.
Las líneas 6610 a 6600 comprueba la existencia del Z$ en el microdrive y actualiza las variables de control.
La línea 6690 posibilita la grabación del archivo.
La comprobación sobre el espacio.
La línea 6690 graba el archivo.
Si el nombre dado ya existiese en el cartucho, se preguntará si se produce la actualización de la información contenida bajo el mismo nombre (6800), y si es así se elimina el archivo anterior (6850) y se graba en el nuevo archivo.

Uno de los mayores placeres de cuantos nos dedicamos al periodismo, es poder escribir los artículos con un buen editor de textos. Por eso, cuando empezamos a trabajar con el Tasword, *en seguida quedamos prendados de la versatilidad con que nos permitía escribir. Cuando diseñamos un* interface *para la impresora, con el que trabajar de forma profesional, el ordenador empezó a mostrar un funcionamiento anómalo... ¡El editor había encontrado su compañera ideal!*



Cuando al editor le conquistó la impresora

En el número anterior se describió la construcción de un económico *interface* CENTRONICS para impresora. El objetivo de este artículo es explicar las modificaciones que habremos de hacer en el **Tasword,** editor de textos comentado igualmente en el anterior número, para poder usarlo con el mencionado *interface.*

Dado que la información que se ofrece con el **Tasword** (o Context en la versión española), sobre los códigos de control de los *interfaces* es a todas luces insuficiente para nuestros propósitos, se hacía necesario la realización de un programa que gestionase la salida de textos a impresora. La ventaja que esto supone es la de permitirnos la obtención de hojas impresas más acordes con nuestro gusto y, de paso, solucionar el siempre presente problema de las eñes. La ñ no existe en el inglés, por ello resulta muy difícil encontrar eñes en los teclados de los ordenadores.

Para imprimir la "ñ", haremos uso de las posibilidades gráficas que ofrecen las impresoras de matriz de puntos. Se puede decir que vamos a hacer un dibujo de la letra. Siempre que queramos que aparezca una ñ habremos de utilizar algún caracter no usado. No haremos que aparezca en pantalla, pero podremos estar seguros de que aparecerá impreso correctamente en el papel. En este programa se ha usado la @, ya que apenas se utiliza. Solamente hemos definido la ñ minúscula porque una pantalla llena de @ y # podría dar lugar a muchos problemas de interpretación. En cualquier caso la modificación que habría que hacer al programa no sería muy problemática.

El archivo de texto

En primer lugar, es fundamental conocer cómo el **Tasword** dis-



tribuye la información escrita, es decir, de qué manera el texto se estructura dentro de la memoria del ordenador. Afortunadamente para nosotros es extremadamente sencillo.

La dirección de comienzo de la zona reservada al texto *('text buffer')* puede conocerse fácilmente haciendo PRINT PEEK 62216 + 256*PEEK 22217 y da el valor 32000. Cuando estamos trabajando en BASIC y queremos conocer la longitud del texto, nos basta con escribir PRINT A.

Si con estos datos investigamos en el que hemos llamado *text buffer* y tenemos en cuenta que una línea está compuesta de 64 caracteres, llegaremos a las siguientes conclusiones:

—El texto se almacena en memoria en forma literal. Por ejemplo, a una línea en blanco corresponden 64 bytes con el valor de 32, valor del espacio según el código ASCII. Si se usan márgenes, las bandas laterales vacías se rellenan de espacios. Los símbolos gráficos están representados por el valor correspondiente según el juego de caracteres que usa el Spectrum.

—No existen códigos de control (fin de línea, salto de página, etc....).

De esta manera, para imprimir un texto habremos de mandar a la impresora los números que encontremos en el *text buffer*, en bloques de 64, y a continuación dar la orden de imprimir y avanzar el papel. La única excepción lo constituyen:

—Los caracteres gráficos: daremos al *interface* unos valores diferentes según el significado acordado para cada carácter. Por ejemplo si encontramos un **L**, que significa que debemos empezar a escribir caracteres a doble anchura, la impresora debe recibir el número 0Eh, (14 en decimal, cuyo mnemónico es SO,*Shift Out*).

—La@: en vez de mandar el número 64 (que supondría imprimir dicho símbolo), debemos pasar al modo gráfico (enviamos los códigos de ESC K (27,75) y a continuación, indicamos el número de puntos que va a tener el gráfico a lo ancho (a lo alto es invariablemente 8).

El programa en ENSAMBLADOR

En la figura 1 se ve un organigrama simplificado del programa.

La impresora usada, la MX-70, cuenta únicamente con la 'exquisitez' de caracteres a doble anchura. Para impresoras más modernas podemos hacer subrayados, caracteres en cursiva,...

Las posibles modificaciones al programa de control son sencillas y se explican más adelante.

El programa definitivo muestra la estructura que se plantea en la fig. 2.

Se ha preferido que la cabecera de la hoja se imprima desde el BASIC ya que es este el lugar que más "toques personales" puede recibir y en consecuencia debe poder cambiarse fácilmente.

Así, es que, cuando se ha impreso una hoja, el control del programa pasa al intérprete BASIC que se encarga de imprimir el número de la página y de dejar las líneas en blanco necesarias.

Por ello siempre que volvamos



Si le interesan los montajes y realizó el que le proponíamos en el número anterior sobre el *interface* CENTRONICS, esta es su oportunidad de incorporar un poco de software para sacarle el máximo rendimiento.

al BASIC será necesario guardar la dirección del carácter próximo a escribir. El lugar donde se guarda este número es lo que hemos llamado HLSAVE. Después de estos dos *bytes* de reserva, tenemos la subrutina PRINT, la encargada de gestionar el intercambio de información con el *interface* (la burocracia). Es idéntico al que se explicó en el artículo dedicado al *interface*, por lo que no requiere más comentarios.

A continuación encontramos el conjunto de números que hay que pasar al *interface* para que escriba una ñ. Lo hemos llamado ENNEG por aquello de Gráficos de la ENNE.

El programa 1 se usará para introducir en HLSAVE el valor de la dirección de comienzo del texto. En consecuencia solo se utilizará una vez al empezar a imprimir un documento.

El segundo programa será el de uso general. Se pasa a describir a continuación, en la fig. 3. Paralelamente ofrecemos un diagrama de flujo ya más detallado.

El registro C se ha usado como contador de las líneas que quedan por imprimir. El valor que toma inicialmente podrá cambiarse (ver programa en BASIC). Después de inicializar la C, se "imprime" un número de espacios correspondientes al margen izquierdo. También el número de estos espacios puede alterarse.

El registro B contendrá el número de columnas por página (64 en el **Tasword**). Se hace uso de la cómoda instrucción DJNZ.A continuación se observa si el carácter a imprimir tiene un código por debajo de 32 (esto debería ser imposible), en cuyo caso volveremos al BASIC para informar del error. Bien podría ocurrir que tuviéramos caracteres gráficos. Si así fuera comprobamos si se trata de alguno de los dos que funcionan con nuestra impresora. En caso de usarse otra que admitiera otros caracteres de control, podremos ampliar el programa en la misma filosofía, esto es:

```
      LD A, Código del gráfico
      CP (HL)
      JR NZ, OTRO
      LD A, Código impresora
      JR ENTRY1
OTRO  LD A,...
```

Otra posibilidad es que nos encontramos con la @. En este caso enviaríamos a la impresora los da-

PROGRAMA 1

```
 200 REM Imprimir texto
 210 CLS : GO SUB 4000
 220 LET A$="D3FF3E80D3DFDBDFE60
128FAC9184B0500BEA0A01E002A08F32
2005BC92A005B0E3C06083E20CD025B1
0F906403E1FBE301D3E7EBE38423E40B
E282D7ECD025B23"
 230 LET A$=A$+"10EA3E0ACD025B3E
0FDBFE1F380422005BC93E00BC300720
F57DFE0038F00D20C03E0CCD025B18E6
D90609110F5B1ACD025B1310F9D918C7
3E8EBE20043E0E18BB3E81BE20C83E14
18B2"
 240 FOR I=1 TO LEN A$/2: POKE 2
3297+I,16*FN H(A$(2*I-1))+FN H(A
$(2*I)): NEXT I
 250 DEF FN H(W$)=CODE W$-CODE "
0"-7*(W$>"9")
 260 INPUT "Comienzo columna (8)
";A$: IF A$="" THEN LET A$="8"
 262 POKE 23333,VAL A$
 264 INPUT "Numero de lineas por
 pagina (60)";A$: IF A$="" THEN
LET A$="60"
 266 LET LIN=VAL A$: POKE 23331,
LIN: LET ESP=INT (720/LIN)
 268 INPUT "Separacion entre lin
eas"'" NORMAL:12"'(" RECOMEND.("
;ESP;")")'A$: IF A$="" THEN LET
A$=STR$ ESP
 270 LET GUARD=A: LET A=27: GO S
UB 400: LET A=65: GO SUB 400: LE
T A=VAL A$: GO SUB 400
 272 INPUT "Comienzo en pagina (
1)";A$: IF A$="" THEN LET A$="1"
 274 LET PAG=VAL A$-1
 280 PRINT AT 20,0; FLASH 1;"
Apriete SPACE para parar   "
 285 LET B=FN P(62216)+GUARD: PO
KE 23382,INT (B/256): POKE 23390
,B-256*INT (B/256): RANDOMIZE US
R 23320
 290 FOR I=1 TO GUARD/64/LIN+1
 300 LET A=10: GO SUB 400
 310 FOR J=1 TO 32: LET A=CODE "
": GO SUB 400: NEXT J: LET A=CO
DE "-": GO SUB 400: LET A$=STR$
(I+PAG): FOR J=1 TO LEN A$: LET
A=CODE A$(J): GO SUB 400: NEXT J
: LET A=CODE "-": GO SUB 400: LE
T A=10: GO SUB 400: LET A=10: GO
 SUB 400
 320 IF NOT USR 23327 THEN NEXT
I: LET A=GUARD: GO TO 25
 330 IF I<>INT (GUARD/64/LIN+1)
THEN CLS : BEEP .1,30: PRINT AT
10,13; FLASH 1;"ERROR": STOP
 340 LET a=GUARD: GO TO 25
 400 OUT 255,A: OUT 223,128
 410 LET A=IN 223: IF A=2*INT (A
/2) THEN GO TO 410
 420 RETURN
```

Del 16 al 23 de Noviembre de 1984

ESTAREMOS EN EL SIMO'84



Stand-D-168
PABELLON XI

Stand-E-14
PABELLON XII

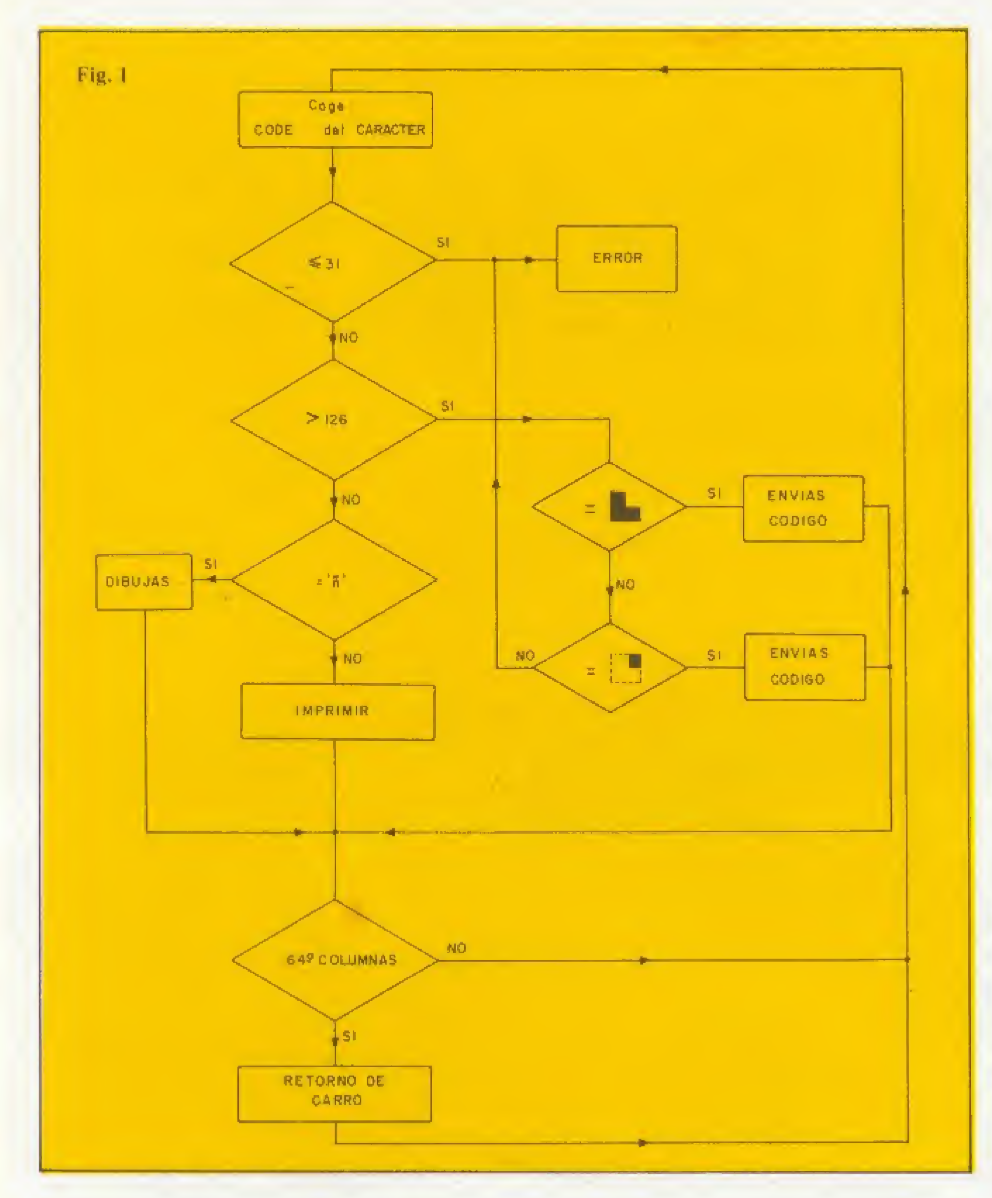


Fig. 2

```
5B00 }
5B01 } HLSAVE
5B02 }
     } Subrut. PRINT
5B0E
5B0F }
     } ENNEG
5B17
5B18 }
     } PROG1
5B1E
5B1F }
     } PROG2
5B8C
```

tos correspondientes a la ñ. Si quisiéramos definir la ñ mayúscula, deberíamos comparar de nuevo si se trata del carácter correspondiente, y enviar los números adecuados.

Si no ocurre ninguna de estas situaciones, se imprime el carácter. Comprobamos si hemos acabado la línea. En este caso, daríamos a la impresora el código de imprimir y avanzar el papel. A continuación muestreamos la tecla BREAK. Si se estuviera apretando, volveríamos al BASIC. En caso contrario, comparamos la dirección de la le-

PROGRAMA 2

```
;  ********************
;  **** TASWORD II ****
;  ********************

;  © Luis Miguel BRUGAROLAS


FF      EQU  12;(Form Feed),salto
;            de pagina
LF      EQU  10;(Line Feed),
;            imprimir linea y avanzar
;            papel


HLSAVE  DEFS 2

PRINT   OUT  (255),A
        LD   A,80H
        OUT  (223),A
LOOP    IN   A,(223)
        AND  1
        JR   Z,LOOP
        RET

ENNEG   DEFB 25,'K',5,0
;       Codigo de ESC K,<n2>,<n1>
;       donde LONG GRAF=256*n1+n2
        DEFB 10111110B
        DEFB 10100000B
        DEFB 10100000B
        DEFB 00111110B
        DEFB 00000000B

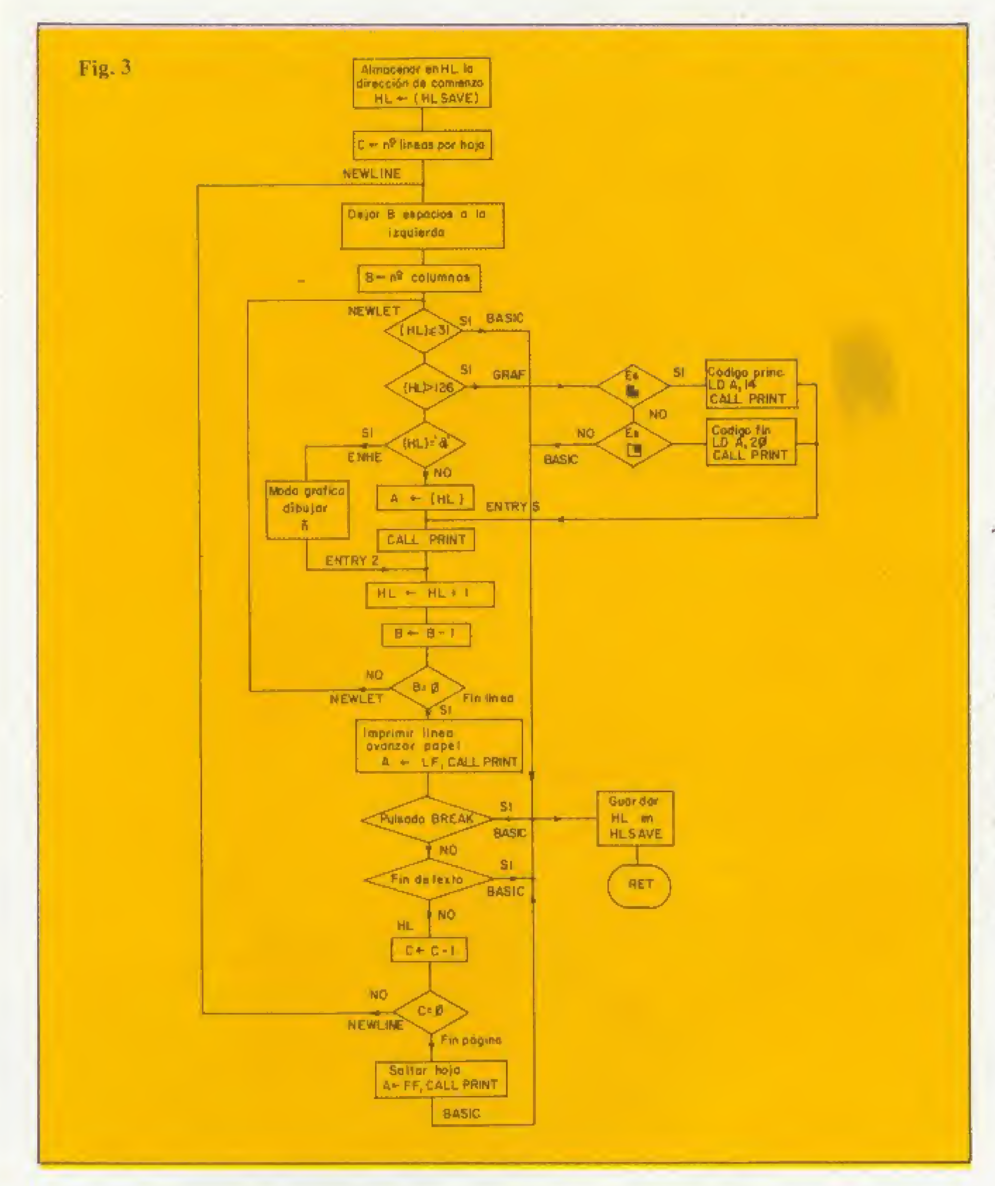
PROG1   LD   HL,(62216)
        LD   (HLSAVE),HL
        RET

PROG2   LD   HL,(HLSAVE)
        LD   C,60;Num lineas/hoja

NEWLIN  LD   B,8;Num espacio/hoja
        LD   A,' '
        CALL PRINT
        DJNZ MARGEN

        LD   B,64;Num column/hoja
NEWLET  LD   A,31
        CP   (HL)
        JR   NC,BASIC;ERROR,cod.
;       no valido
        LD   A,126
        CP   (HL)
        JR   C,GRAF;Tal vez GRAF
```

Fig. 3



tra sobre la que trabajamos con la de final del texto. Si coincidieran, volveríamos al menú del **Tasword.** En otro caso, continuaríamos hasta acabar con la página. Entonces volvemos al BASIC, esta vez para imprimir la cabecera, y de nuevo pasamos el control de la operación a este programa.

Modificaciones al Tasword

Si bien es muy interesante conocer el programa anterior, no es en absoluto imprescindible para poder usarlo. A continuación vamos a explicar los cambios que tenemos que hacer al **Tasword** para poder usarlo con nuestro *interface.*

—Cargar el **Tasword** en el ordenador.

—Borrar todas las líneas de BASIC entre 200 y 490 (inclusive).

—Eliminar las líneas 45, 160, 660.

—Modificar las siguientes líneas:

```
20 INK 7:PAPER 0:BORDER 0:CLS:LET A=...
125 IF B=116 THEN LET I=14
170 IF B=98 THEN LET I=16
610 IF B=121 THEN INK 0:PAPER 7:CLS:GOTO 10
```

—Escribir el programa 1, y com-

```
        LD   A,'@'
        CP   (HL)
        JR   Z,ENHE;Es la enhe
        LD   A,(HL)
ENTRY1  CALL PRINT
ENTRY2  INC  HL
        DJNZ NEWLET

        LD   A,LF;Imprimir
        CALL PRINT

;Volver a BASIC si se ha pulsado
;BREAK
        LD   A,0FH
        IN   A,(0FEH)
        RRA
        JR   C,CONT

BASIC   LD   (HLSAVE),HL
        RET

CONT    LD   A,0;Hacer  POKE con
;       parte alta direccion fin
        CP   H
        JR   NC,NL
        JR   NZ,BASIC
        LD   A,L
        CP   0;Hacer POKE con
;       parte baja direccion fin
        JR   C,BASIC
NL      DEC  C
        JR   NZ,NEWLIN
        LD   A,FF;Salto de pag
        CALL PRINT
        JR   BASIC

ENHE    EXX
        LD   B,09
        LD   DE,ENNEG
DOTS    LD   A,(DE)
        CALL PRINT
        INC  DE
        DJNZ DOTS
        EXX
        JR   ENTRY2

GRAF    LD   A,142
        CP   (HL)
        JR   NZ,NOABR
        LD   A,14;Cod car DOBLES
        JR   ENTRY1
NOABR   LD   A,129
        CP   (HL)
        JR   NZ,BASIC
        LD   A,20;Cod caract NORM
        JR   ENTRY1
```

probar que está correctamente escrito.

—Grabar el programa en BASIC en cinta, y comprobar.

—Descansar por el esfuerzo realizado.

—Cargar el programa en BASIC grabado previamente y después el programa en código máquina original.

—Grabar de nuevo.

Tal vez pueda parecer algo raro los últimos pasos pero pueden surgir problemas debidos a la falta de memoria si no se siguen correctamente.

ESCUELA DE COCINA PANTAGRUEL

COCINA SELECTA

Lepanto 56
28081 MADRID

Dña Cristina Caballero
Americas 91
28091 MADRID

2-Octubre-1984

Estimada señora:

Ha resultado usted admitida en nuestra prestigiosa casa para cursar sus estudios de cocina.

Las clases comenzaran el Lunes 19 de Noviembre en nuestros locales. Con objeto de preparar las clases practicas, sirvase traer los siguientes alimentos:

-Un Kilogramo de chuletas de cerdo
-Dos paquetes de harina
-Una docena de huevos frescos
-Un litro de aceite de oliva

La esperamos a las 9 de la mañana.
Queda de usted afectisimo:

Luis Hernandez Ruiz
Jefe de Servicios Culinarios.

Ejemplo de impresión. Para escribir cartas comerciales se evitará el paginado eliminando las líneas 272, 274 y 310.

Funcionamiento

En cuanto al funcionamiento del programa solo hay que decir que en las opciones que se piden, si queremos el valor entre paréntesis es suficiente con apretar ENTER. Si resulta muy latoso dar todos los datos se pueden eliminar los INPUTS que cada uno considere necesario, o bien cambiar los valores por defecto.

Una última anotación. Al seleccionar la opción de impresión se observa que la pantalla aparece negra algún tiempo. Mientras tanto se introduce el código máquina en el *buffer* de impresora. Ha de realizarse cada vez para estar seguros de que va a continuar en este sitio ya que otras opciones del **Tasword** usan la misma zona de memoria. Con un DATA la transferencia es mucho más rápida, pero ocupa mucha más memoria...

Luis Miguel

Preguntas y respuestas

P En el programa Easel del número dos no reconocemos los caracteres empleados en las instrucciones POKE de la línea 50. ¿Es un fallo de impresión?

Luis Matienzo
Madrid

R Se trata de caracteres definidos por el usuario (UDG). Repetimos esta instrucción previamente a su definición, donde puede verse que los caracteres empleados se puede acceder fácilmente presionando A, B, y C en modo gráfico.

```
  40 POKE 23658,0: FOR f=USR "a"
 TO USR "c"+7: POKE f,0: NEXT f
  50 POKE USR "A"+4,255: POKE US
R "B"+4,204: POKE USR "C"+4,240:
 LET k$="ABC"


 A B C D E ...
 — ·· - D E ...
```

P Veo con agrado en el número 2, un esbozo de programas en FORTH y PASCAL. Los «científicos de por libre» como es mi caso veríamos con mucho agrado una sección dedicada al FORTH y PASCAL.

José Perona
Murcia

R En este mismo número hacemos una breve introducción al FORTH. Una serie de artículos sobre este tema comienza precisamente en este mismo mes en la revista ZX. Si está interesado en el PASCAL, esperamos verá con agrado la serie que abrimos sobre este lenguaje en el próximo número de TODOSPECTRUM.



Frogger

Con 48K, un poco de habilidad y algún que otro susto podrá llevar a casa a su rana favorita. Para ello ha de evitar ser atropellado en la carretera, teniendo en cuenta que a partir de la tercera fila de coches sólo puede desplazarse hacia adelante o hacia atrás, pero no puede moverse horizontalmente, lo cual le añade una mayor complejidad. Una vez que se llega a la parte central podrá tomarse un poco de respiro antes de meterse a la aventura acuática. Sobre el río van unos troncos y dos filas de tortugas traviesas, especialmente la segunda, que les gusta sumergirse de vez en cuando. Lógicamente ha de alcanzar el extremo de la pantalla saltando sobre las tortugas y troncos, lo cual no resulta fácil.

Los gráficos están muy bien logrados y es bastante rápido, resultando difícil de creer que esté realizado en BASIC. Otro aspecto de destacar es su buena estructuración:

Líneas 10-20

Inicialización del programa (s = puntuación, li = vidas y h() es la tabla de records).

Líneas 30-50

Inicialización posterior (fr = casa de la rana, t = tiempo restante, "a" y "b" = coordenadas de la rana).

Líneas 60-67

Comienzo del bucle principal. Se reduce el tiempo (t), impri-

17




```
   5 REM **************
         *            *
         *  FROGGER   *
         *            *
         **************
  10 DIM h(6): GO SUB 9000: REM
Comienzo
  20 LET s=0: LET li=3:
  30 GO SUB 7500: REM Nivel
  35 LET fr=0
  40 GO SUB 7000: REM Pantalla
  50 LET t=100: LET a=21: LET b=
16: PRINT AT a,b; INK 4;"■"
  60 REM Bucle principal
  65 BEEP .005,-15
  66 LET t=t-1: PRINT AT 0,18; I
NK 0; PAPER 5;t;" "
  67 IF t<=0 THEN GO TO 3000
  70 LET a$=a$(32)+a$( TO 31): L
ET h$=h$(32)+h$( TO 31)
  80 LET b$=b$(2 TO )+b$(1): LET
 e$=e$(2 TO )+e$(1): LET g$=g$(2
 TO )+g$(1)
  90 PRINT PAPER 5;AT 3,0; INK 0
;a$;AT 5,0; INK 2;b$;AT 7,0; INK
 0;a$;AT 9,0; INK 2;b$
  95 BEEP .005,-15
 100 PRINT PAPER 0;AT 13,0; INK
7;e$;AT 15,0; INK 2;h$;AT 17,0;
INK 3;g$;AT 19,0; INK 2;h$
 110 IF a=3 OR a=7 THEN LET b=b+
1: IF b=31 THEN GO TO 3000
 115 IF a=15 OR a=9 THEN LET b=b
-1: IF b=0 THEN GO TO 3000
 120 GO SUB 200
 130 IF INKEY$="7" THEN GO TO 10
00
 140 IF INKEY$="8" OR INKEY$="5"
 THEN GO SUB 300
 150 IF a=3 OR a=7 OR (a>11 AND
a<21) THEN PAPER 0
 151 IF a=5 OR a=9 THEN PAPER 2
 152 IF a=11 OR a=21 THEN PAPER
1
 153 PRINT AT a,b; INK 4;"■"
 160 IF INKEY$="6" THEN GO SUB 1
500
 180 GO TO 60
 200 IF a=21 OR a=11 THEN RETURN

 201 GO TO 200+a
 202 RETURN
 203 IF a$(b+1)=" " THEN GO TO 3
000
 204 GO TO 202
```

Programas

miéndose su valor en la pantalla. La línea 67 comprueba si ha finalizado el tiempo.

Líneas 70-120

Rutina de movimiento. Las líneas 70 y 80 actualizan los valores del *string* para simular el movimiento de izquierda a derecha y de derecha a izquierda. Las líneas 90 a 100 imprimen los objetos en sus nuevas posiciones. Las líneas 110 y 115 verifican si la rana se encuentra encima de una tortuga o encima de un tronco.

Líneas 130-180

Rutina de movimiento de la rana. Comprueba si se ha presionado una de las teclas del cursor para desplazamientos a la izquierda, derecha, arriba o abajo. El color del PAPER se activa dependiendo de la posición de la rana.

Líneas 200-219

Rutina de verificación de la rana. Detecta si ha chocado con un coche o un camión o si ha caído al agua, en cuyo caso se pasa el control a la rutina de muerte de la rana.

Líneas 300-330

Rutina que comprueba si la rana todavía se encuentra en la pantalla, incrementando o decrementando "b" si el movimiento es hacia la izquierda o derecha respectivamente. El color de PAPER se activa dependiendo de la posición de la rana y la línea 320 imprime la rana en la nueva posición.

Líneas 1000-1090

Rutina de movimiento ascendente de la rana. La línea 1015 comprueba si se ha llegado al final de la pantalla. La línea 1017 llama a la rutina de "choque con obje-



17

```
 205 IF b$(b+1)=" " THEN GO TO 3
000
 206 GO TO 202
 207 GO TO 203
 209 GO TO 205
 213 IF e$(b+1)<>" " THEN GO TO
3000
 214 GO TO 202
 215 IF h$(b+1)<>" " THEN GO TO
3000
 216 GO TO 202
 217 IF g$(b+1)<>" " THEN GO TO
3000
 218 RETURN
 219 GO TO 215
 300 IF a=3 OR a=7 OR (a>11 AND
a<21) THEN PAPER 0
 301 IF a=5 OR a=9 THEN PAPER 2
 302 IF a=11 OR a=21 THEN PAPER
1
 303 PRINT AT a,b;" "
 310 BEEP .01,10: LET b=b+(INKEY
$="8" AND b<31)-(INKEY$="5" AND
b>0)
 320 PRINT AT a,b; INK 4; PAPER
1; PAPER 0 AND a>11 AND a<21; PA
PER 5 AND a<11;"■"
 330 RETURN
1000 IF a=3 OR a=7 OR (a>11 AND
a<21) THEN PAPER 0
1001 IF a=5 OR a=9 THEN PAPER 2
1002 IF a=11 OR a=21 THEN PAPER
1
1003 PRINT AT a,b;" "
1010 LET a=a-2: BEEP .05,10
1015 IF a=1 THEN GO TO 2000
1017 GO SUB 200
1050 PRINT AT a,b; INK 4; PAPER
1; PAPER 0 AND a>11 AND a<21; PA
PER 5 AND a<11;"■"
1060 LET s=s+10: PRINT AT 0,15-L
EN STR$ s; PAPER 5; INK 0;s
1090 GO TO 60
1500 IF a=3 OR a=7 OR (a>11 AND
a<21) THEN PAPER 0
1501 IF a=5 OR a=9 THEN PAPER 2
1502 IF a=11 OR a=21 THEN PAPER
1
1503 PRINT AT a,b;" "
1505 IF a=21 THEN RETURN
1510 LET a=a+2: BEEP .05,10
1540 GO SUB 200
1550 PRINT AT a,b; INK 4; PAPER
1; PAPER 0 AND a>11 AND a<21; PA
PER 5 AND a<11;"■"
1590 GO TO 60
2000 IF b<>2 AND b<>8 AND b<>16
AND b<>23 AND b<>29 THEN GO TO 3
000
2001 IF ATTR (1,b)=44 THEN GO TO
 3000
2002 LET s=s+100: PRINT AT 0,15-
LEN STR$ s; PAPER 5; INK 0;s
2003 PRINT AT 1,b; INK 4; PAPER
5;"■"
2005 LET fr=fr+1: BEEP .05,15: B
EEP .05,20
2010 IF fr=5 THEN GO TO 2200
2020 GO TO 50
2200 PRINT AT 11,10; INK 7; BRIG
HT 1; PAPER 1;"Punt. ";t*10
2210 FOR f=0 TO 3: BEEP .1,2: BE
EP .1,4: BEEP .1,6: BEEP .1,6: B
EEP .1,4: BEEP .1,0: NEXT f: LET
 s=s+1000
2215 LET s=s+(t*10): PRINT AT 0,
15-LEN STR$ s; PAPER 5; INK 0;s
2220 FOR f=1 TO 5: FOR n=1 TO 5:
 BEEP .01,f+n: NEXT n: NEXT f: G
O TO 30
```

tos". La línea 1050 imprime la rana en la nueva posición, y la línea 1060 incrementa e imprime la puntuación.

Líneas 1500-1590

Rutina de movimiento descendente. Es análoga a la anterior, con la diferencia de que no existe verificación de la llegada a la parte superior de la pantalla.

Líneas 2000-2020

Rutina de comprobación de estar en casa. La línea 2000 comprueba si se alcanzó una posición correcta (un hueco) al llegar a al extremo superior de la pantalla. La línea 2001 comprueba si todas las posiciones llegada están llenas y la 2010 comprueba si todas las ranas han llegado a las posiciones finales.

Líneas 2200-2220

Rutina que entra en funcionamiento cuando todas las ranas han llegado a casa. La línea 2200 imprime el incremento de tiempo conseguido y la línea 2210 hace sonar una pequeña melodía e incrementa la puntuación.

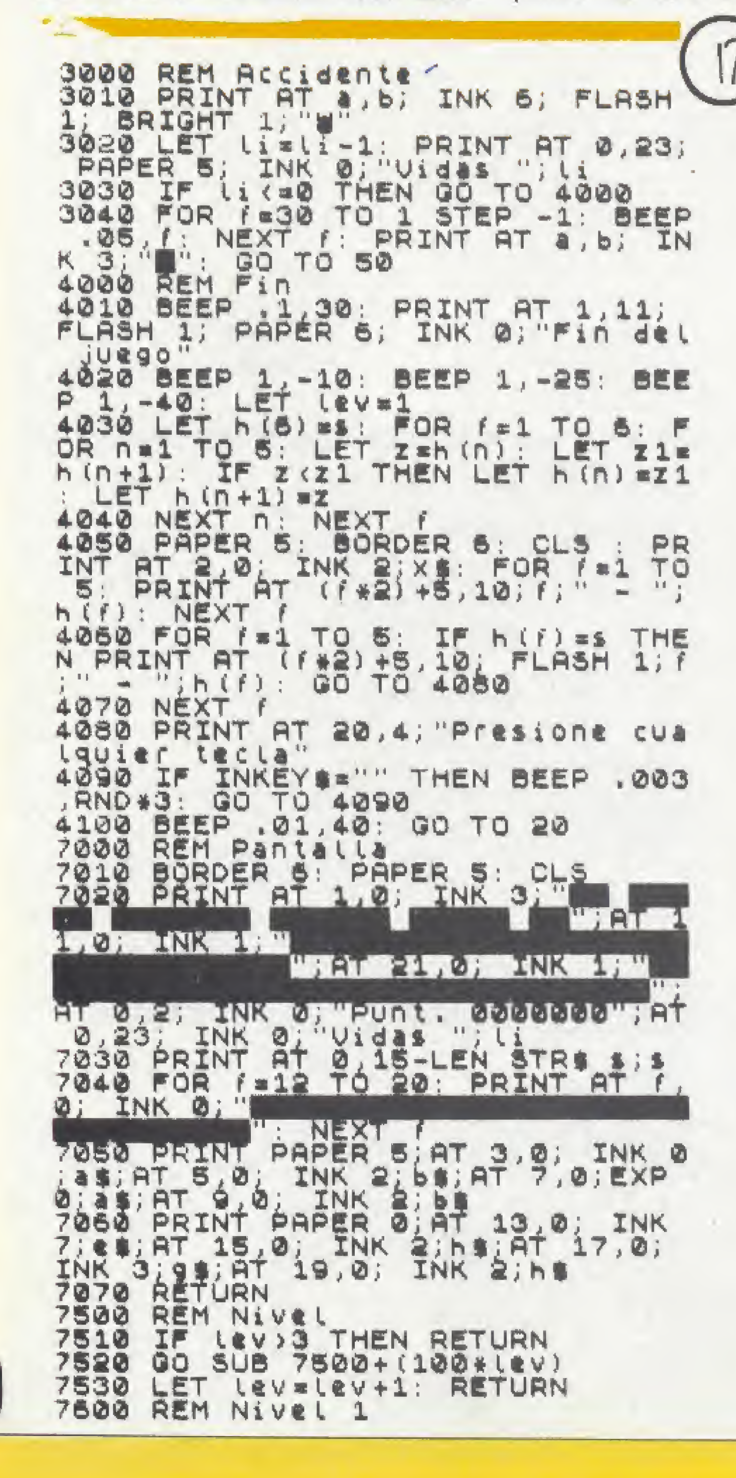
Líneas 3000-3040

Rutina de muerte de la rana. La línea 3010 imprime una rana intermitente y la 3020 imprime las vidas que quedan. La línea 3030 comprueba si existen vidas.

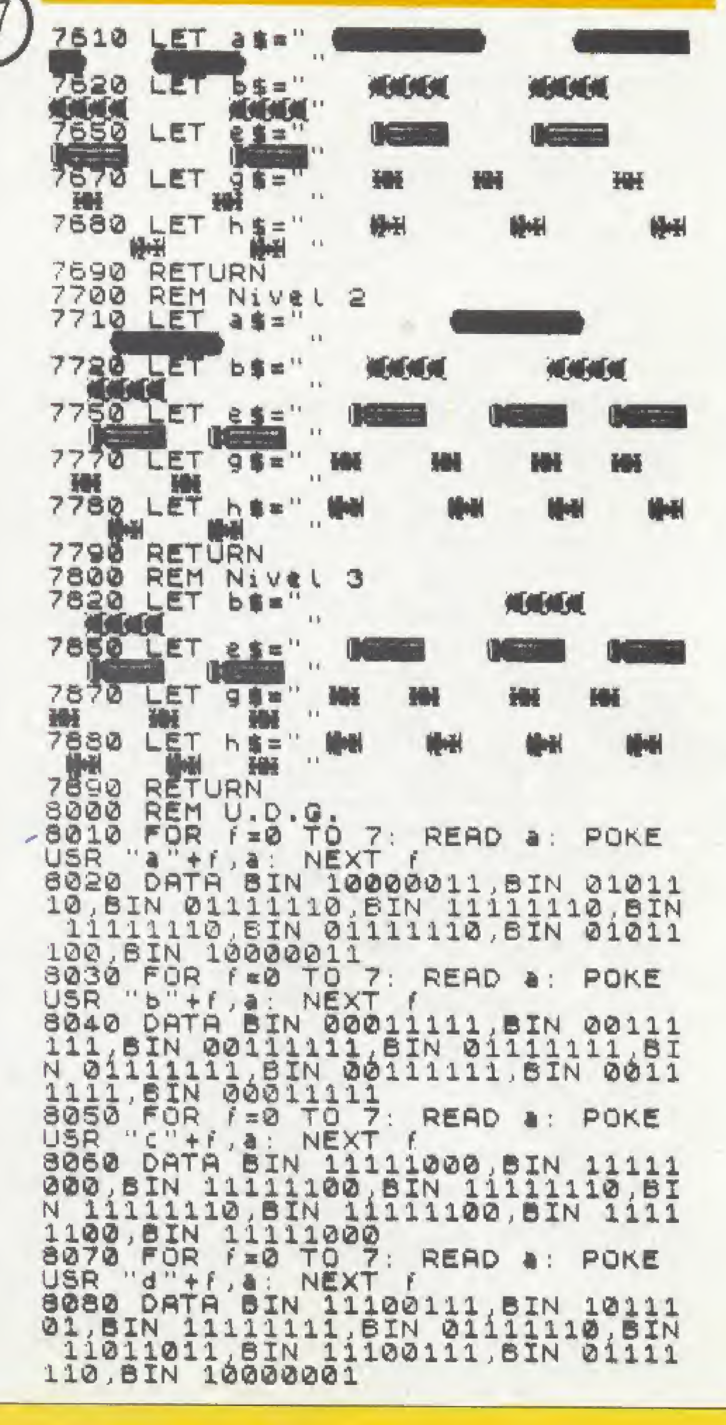
Línas 4000-4100

Rutina de fin de programa. La línea 4020 realiza un reset del nivel de dificultad. Las líneas 4030 y

17



```
3000 REM Accidente
3010 PRINT AT a,b; INK 6; FLASH 1; BRIGHT 1;"■"
3020 LET li=li-1: PRINT AT 0,23; PAPER 5; INK 0;"Vidas ";li
3030 IF li<=0 THEN GO TO 4000
3040 FOR f=30 TO 1 STEP -1: BEEP .05,f: NEXT f: PRINT AT a,b; INK 3;"■": GO TO 50
4000 REM Fin
4010 BEEP .1,30: PRINT AT 1,11; FLASH 1; PAPER 6; INK 0;"Fin del juego"
4020 BEEP 1,-10: BEEP 1,-25: BEEP 1,-40: LET lev=1
4030 LET h(6)=s: FOR f=1 TO 5: FOR n=1 TO 5: LET z=h(n): LET z1=h(n+1): IF z<z1 THEN LET h(n)=z1: LET h(n+1)=z
4040 NEXT n: NEXT f
4050 PAPER 5: BORDER 6: CLS : PRINT AT 2,0; INK 2;x$: FOR f=1 TO 5: PRINT AT (f*2)+5,10;f;" - ";h(f): NEXT f
4060 FOR f=1 TO 5: IF h(f)=s THEN PRINT AT (f*2)+5,10; FLASH 1;f;" - ";h(f): GO TO 4080
4070 NEXT f
4080 PRINT AT 20,4;"Presione cualquier tecla"
4090 IF INKEY$="" THEN BEEP .003,RND*3: GO TO 4090
4100 BEEP .01,40: GO TO 20
7000 REM Pantalla
7010 BORDER 6: PAPER 5: CLS
7020 PRINT AT 1,0; INK 3;"██████████";AT 11,0; INK 1;"██████████";AT 21,0; INK 1;"██████████";AT 0,2; INK 0;"Punt. 0000000";AT 0,23; INK 0;"Vidas ";li
7030 PRINT AT 0,15-LEN STR$ s;s
7040 FOR f=12 TO 20: PRINT AT f,0; INK 0;"██████████": NEXT f
7050 PRINT PAPER 5;AT 3,0; INK 0;a$;AT 5,0; INK 2;b$;AT 7,0;EXP 0;a$;AT 9,0; INK 2;b$
7060 PRINT PAPER 0;AT 13,0; INK 7;e$;AT 15,0; INK 2;h$;AT 17,0; INK 3;g$;AT 19,0; INK 2;h$
7070 RETURN
7500 REM Nivel
7510 IF lev>3 THEN RETURN
7520 GO SUB 7500+(100*lev)
7530 LET lev=lev+1: RETURN
7600 REM Nivel 1
```



```
7610 LET a$="  ████   ████  ██  "
7620 LET b$="  ▓▓▓▓   ▓▓▓▓  ▓▓▓▓     ▓▓▓▓"
7650 LET e$="  ▐██▌   ▐██▌  ▐██▌   ▐██▌"
7670 LET g$="  ▪▪   ▪▪    ▪▪   ▪▪   ▪▪"
7680 LET h$="  ▪▪   ▪▪    ▪▪   ▪▪   ▪▪"
7690 RETURN
7700 REM Nivel 2
7710 LET a$="    ████   ████  "
7720 LET b$="  ▓▓▓▓   ▓▓▓▓  ▓▓▓▓  "
7750 LET e$=" ▐██▌  ▐██▌  ▐██▌  ▐██▌  ▐██▌ "
7770 LET g$=" ▪▪  ▪▪  ▪▪  ▪▪  ▪▪  ▪▪ "
7780 LET h$=" ▪▪  ▪▪  ▪▪  ▪▪  ▪▪  ▪▪ "
7790 RETURN
7800 REM Nivel 3
7820 LET b$="      ▓▓▓▓   ▓▓▓▓  "
7850 LET e$=" ▐██▌  ▐██▌  ▐██▌  ▐██▌  ▐██▌ "
7870 LET g$=" ▪▪  ▪▪  ▪▪  ▪▪  ▪▪  ▪▪  ▪▪ "
7880 LET h$=" ▪▪  ▪▪  ▪▪  ▪▪  ▪▪  ▪▪  ▪▪ "
7890 RETURN
8000 REM U.D.G.
8010 FOR f=0 TO 7: READ a: POKE USR "a"+f,a: NEXT f
8020 DATA BIN 10000011,BIN 01011110,BIN 01111110,BIN 11111110,BIN 11111110,BIN 01111110,BIN 01011100,BIN 10000011
8030 FOR f=0 TO 7: READ a: POKE USR "b"+f,a: NEXT f
8040 DATA BIN 00011111,BIN 00111111,BIN 00111111,BIN 01111111,BIN 01111111,BIN 00111111,BIN 00111111,BIN 00011111
8050 FOR f=0 TO 7: READ a: POKE USR "c"+f,a: NEXT f
8060 DATA BIN 11111000,BIN 11111000,BIN 11111100,BIN 11111110,BIN 11111110,BIN 11111100,BIN 11111100,BIN 11111000
8070 FOR f=0 TO 7: READ a: POKE USR "d"+f,a: NEXT f
8080 DATA BIN 11100111,BIN 10111101,BIN 11111111,BIN 01111110,BIN 11011011,BIN 11100111,BIN 01111110,BIN 10000001
```

4040 actualizan la tabla de records. Las líneas 4050 y 4080 imprimen el título y la tabla de records y la línea 4090 realiza diversos BEEPS en espera de que se presione una tecla.

Líneas 7000-7070

Dibujan la pantalla inicial.

Líneas 7500-7890

Rutina de asignación del número de coches, camiones, troncos y tortugas que aparecerán en la pantalla. La variable "lev" de la línea 7520 origina una bifurcación según el nivel elegido. Si desea añadir un nuevo nivel al juego, puede cambiar el "3" de la línea 7510 por "4". Después añadir las siguientes líneas:

```
7900 REM  LEVEL 4
7910 LET    a$ = "   troncos                  "
7920 LET    b$ = "   tortugas                 "
7930 LET    e$ = "   camiones                 "
7940 LET    g$ = "   coches                   "
7950 LET    h$ = "   coches de carreras       "
7960 RETURN
```

Pueden colocarse todos los caracteres que se deseen, pero las variables habrán de contener siempre 32 caracteres.

Líneas 8000-8500

Rutina de definición de caracteres gráficos.

Líneas 9000-9080

Rutina de inicio de juego. La variable del nivel de juego se inicializará a uno. La línea 9030 imprime el título del juego en grandes caracteres y hace sonar una situación de inicio del juego.

Línea 9090

Rutina de grabación en cassette.

17

```
8090 FOR f=0 TO 7: READ a: POKE
USR "e"+f,a: NEXT f
8100 DATA BIN 0,BIN 01000010,BIN
 01011010,BIN 01111110,BIN 00111
100,BIN 01111110,BIN 01111110,BI
N 00111100
8130 FOR f=0 TO 7: READ a: POKE
USR "g"+f,a: NEXT f
8140 DATA BIN 00111100,BIN 01011
101,BIN 01011111,BIN 01011101,BI
N 01011101,BIN 01011111,BIN 0101
1101,BIN 00111100
8150 FOR f=0 TO 7: READ a: POKE
USR "h"+f,a: NEXT f
8160 DATA BIN 11111111,BIN 11111
111,BIN 0,BIN 11111111,BIN 11111
111,BIN 0,BIN 11111111,BIN 11111
111
8170 FOR f=0 TO 7: READ a: POKE
USR "i"+f,a: NEXT f
8180 DATA BIN 11111111,BIN 11111
111,BIN 00000011,BIN 11111111,BI
N 11111111,BIN 00000011,BIN 1111
1111,BIN 11111111
8190 FOR f=0 TO 7: READ a: POKE
USR "j"+f,a: NEXT f
8200 DATA BIN 00101110,BIN 01101
110,BIN 01110110,BIN 01111111,BI
N 01111111,BIN 01110110,BIN 0110
1100,BIN 00101110
8210 FOR f=0 TO 7: READ a: POKE
USR "k"+f,a: NEXT f
8220 DATA BIN 00000010,BIN 00111
010,BIN 10010010,BIN 11111110,BI
N 11111110,BIN 10010010,BIN 0011
1010,BIN 00000000
8230 FOR f=0 TO 7: READ a: POKE
USR "l"+f,a: NEXT f
8240 DATA BIN 0,BIN 00111011,BIN
 00010010,BIN 00111111,BIN 00111
111,BIN 00010011,BIN 00111011,BI
N 0
8250 FOR f=0 TO 7: READ a: POKE
USR "m"+f,a: NEXT f
8260 DATA BIN 0,BIN 11011100,BIN
 11001000,BIN 11111100,BIN 11111
100,BIN 11001000,BIN 11011100,BI
N 0
8500 RETURN
9000 REM Comienzo
9010 LET lev=1: LET x$="        F
ROGGER             "
9020 BORDER 1: PAPER 6: CLS
9030 PRINT AT 2,0; INK 2;x$: BEE
P .1,10: FOR f=60 TO 30 STEP -1:
 BEEP .05,f: NEXT f: PRINT AT 5,
0; FLASH 1;"
"
9040 PRINT AT 7,11;"PUNTUACION";
AT 9,2;"10 pts. salto adelante."
;AT 11,2;"100 pts. por rana en c
asa.";AT 13,2;"1000 pts. todas l
as ranas en           casa."
9050 PRINT AT 17,1;"Utiliza las
teclas del cursor.": GO SUB 8000
: PRINT AT 20,3; FLASH 1; BRIGHT
 1;"   PULSE CUALQUIER TECLA   "
9060 IF INKEY$="" THEN BEEP .1,R
ND*10+40: GO TO 9060
9070 BEEP .1,2: BEEP .1,4: BEEP
.1,5: BEEP .1,6: BEEP .1,4: BEEP
 .1,0
9080 CLS : RETURN
```

```
7600>REM LEVEL 1
7610 LET a$=" B[redacted]C     B[redacted]C    B[redacted]C   "
7620 LET b$="    AAAA      AAAA
AAAA         AAAA"
7650 LET e$="    GHHI      GHHI
GHHI         GHHI"
7670 LET g$="    LM     LM       LM
 LM      LM    "
7680 LET h$="   JK        JK       JK
     JK      JK  "
7690 RETURN
7700 REM LEVEL 2
7710 LET a$="              B[redacted]C
   B[redacted]C       "
7720 LET b$="    AAAA       AAAA
   AAAA       "
7750 LET e$="  GHHI   GHHI   GHHI
   GHHI  GHHI "
7770 LET g$=" LM   LM    LM   LM
LM    LM      "
7780 LET h$=" JK     JK     JK     JK
    JK      JK  "
7790 RETURN
7800 REM LEVEL 3
7820 LET b$="              AAAA
   AAAA       "
7850 LET e$="  GHHI    GHHI   GHHI
   GHHI  GHHI "
7870 LET g$=" LM   LM    LM   LM
LM    LM    LM "
7880 LET h$=" JK     JK     JK     JK
  JK     JK   LM "
7890 RETURN
```

Detalle de los caracteres gráficos definidos para los tres niveles de juego.



Ariadna

Según la mitología, Ariadna supo salir del laberinto desenrrollando un hilo. De ahí el nombre de este programa-laberinto en el que dispone de dos opciones: ver los pasillos por los que pasa o caminar totalmente a oscuras.

Según el nivel de dificultad elegido (dispone de 10) podrá acceder a los pasadizos. Por supuesto puede colocar cerca el esquema del laberinto, pero el juego no resultaría interesante. Le sugerimos entrar en el pasillo desconocimiento si tendrá salida o tendrá que volver sobre sus pasos. Con 500 movimientos, unos cinco minutos, podrá salir sin dificultad.

No se engañe con los gráficos que le ofrecemos con el listado. La pantalla permanece totalmente oscura, a excepción de la parte derecha donde se encuentra un reloj, un señalizador de movimientos incorrectos y un contador de movimientos. Para ajustar el reloj se ha de cambiar la lína 105 por t = t + (algo más de 6), igualmente la línea 108. Para que la manecilla corta actúe como minutero, las líneas 111 y 113 deben ser modificadas en /12 y /60 respectivamente.

18

```
   2 REM
   3 REM  Laberinto de Ariadna
   4 REM      © Mio    1984
   5 REM
   7 GO SUB 5000
  10 REM
  12 GO SUB 9000
  15 PAPER 0: BORDER 0: INK 0: CLS
  20 GO SUB 4000
  21 LET m=0
  22 INK 7: PLOT 208,0: DRAW 47,0: DRAW 0,175: DRAW -47,0: DRAW 0,-175
  23 PLOT 216,0: DRAW 7,0: DRAW 0,112: DRAW -7,0: DRAW 0,-112
  25 LET x=232: LET y=150: LET s=22: GO SUB 7000
  28 INK 0
  30 FOR n=10 TO ni STEP -1
  31 LET a=INT (RND*20)
  32 LET b=INT (RND*25)
  33 PRINT PAPER 0;AT a,b;CHR$ 32;CHR$ 32
  34 PRINT PAPER 0;AT a+1,b;CHR$ 32;CHR$ 32
  35 NEXT n
  40 INK 0: PLOT 0,0: DRAW 207,0: DRAW 0,175: DRAW -207,0: DRAW 0,-175
  45 PRINT AT 21,13; INK 9; FLASH 1;CHR$ 42
  50 LET x=107: LET y=92
  60 LET t=0
  70 GO TO 100
 100 REM
 105 LET t=t+3
 108 LET t1=t-3
 110 LET a=t/94.24778
 111 LET b=a/12
 112 LET a1=t1/94.24778
 113 LET b1=a1/12
 120 LET sx=18*SIN a
 121 LET sy=18*COS a
 122 LET mx=13*SIN b
 123 LET my=13*COS b
 124 LET sx1=18*SIN a1
 125 LET sy1=18*COS a1
 126 LET mx1=13*SIN b1
 127 LET my1=13*COS b1
 130 IF m>120 THEN GO TO 2000
 141 GO SUB 1000
 145 OVER 1
 150 PLOT 232,150: DRAW sx1,sy1
 151 PLOT 232,150: DRAW mx1,my1
 155 INK 9
 160 PLOT 232,150: DRAW sx,sy
 161 PLOT 232,150: DRAW mx,my
 170 PLOT OVER 0;216,m: DRAW OVER 0;6,0
 200 GO TO 100
1000 REM
1010 LET y1=y: LET x1=x
1020 LET i$=INKEY$
1030 IF i$="q" THEN LET y=y+1
1031 IF i$="z" THEN LET y=y-1
1032 IF i$="p" THEN LET x=x+1
1033 IF i$="i" THEN LET x=x-1
1035 IF i$<>"" THEN LET m=m+.1
1040 IF POINT (x,y)<>0 THEN LET x=x1: LET y=y1: GO SUB 3000
1050 IF y<10 AND x>103 AND x<112 THEN GO SUB 2000
1060 GO SUB tk
1061 PLOT OVER 1; INK 7;x,y
1090 RETURN
2000 REM
2005 PRINT #1; FLASH 1;" Ha logrado salir del laberinto.": PAUSE 0: PAUSE 0
2010 PAPER 7: BORDER 7: INK 0: CLS
2020 GO SUB 4000
2040 FOR n=0 TO 22
2050 PRINT AT 21,31;CHR$ 32;CHR$ 32
2060 NEXT n
2080 INPUT "Desea jugar de nuevo (s/n)? ";n$
2090 IF n$="n" THEN STOP
2100 GO TO 10
3000 REM
3010 LET b$="□NO COMPUTADO□"
3020 FOR n=1 TO 14
3030 PRINT INK 7; BRIGHT 1;AT 6+n,30;b$(n)
3040 NEXT n
3060 FOR n=7 TO 20: PRINT AT n,30;" ": NEXT n
3070 RETURN
4000 REM
4010 RESTORE : FOR n=0 TO 21: READ a$: PRINT a$: NEXT n
4020 RETURN
5000 INK 9: INPUT "Nivel (1 a 10)?";ni
```

El señalizador de movimientos actúa cuando el pixel inmediato al cursor está ocupado por una pared. Hay casos en que esta pared no se percibe en la pantalla por hallarse frente a un pasadizo abierto aleatoriamente.

Los movimientos gastados se visualizan en un control de barras. En principio dispone de 1.200 movimientos, lo que al principio resultará insuficiente. Si le parece excesivo puede alterar la línea 138 por un número mayor de 120 o la 1035 por m = m + .05 para disponer del doble.

El control se realiza mediante Q, Z, P e I.

Análisis del programa

Línea 12

Subrutina 9000. Gráficos definidos.

Líneas 15-100

Preparación de la pantalla.

Líneas 100-200

Bucle principal (funcionamiento reloj).

Línea 141

Salida a subrutina del juego (movimientos, etc.).

Línea 170

Avance del contador de movimiento de barras.

Líneas 1000-1090

INKEY$.

Línea 2000

Salida del laberinto.

Línea 2080

No ha logrado salir del laberinto.

Línea 2090

Fin.

Línea 3000

Control de movimientos inadecuados.

Las líneas 8000 a 8061 se repiten con los caracteres gráficos sin definir para facilitar su introducción.

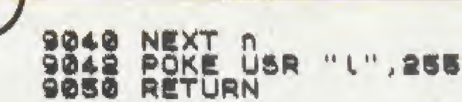
Autor: Igor Beades *16K.*

18

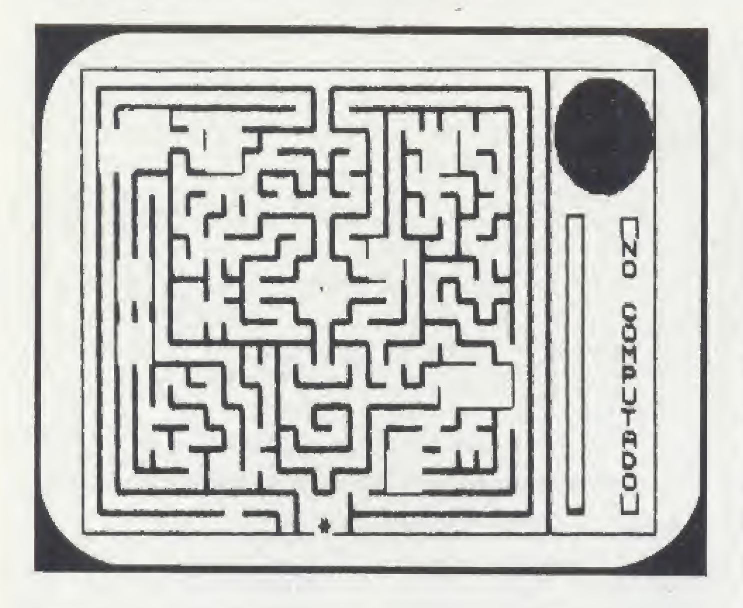
```
5001 IF ni<0 OR ni>10 THEN GO TO
 7
5002 LET ni=INT ni
5007 PRINT AT 5,0;"Desea contar
con la ventaja del hilo <a> o co
nfiar en su sentidode la orienta
cion <p> ?": PAUSE 0
5008 IF INKEY$="p" THEN LET tk=8
050
5009 IF INKEY$<>"p" THEN LET tk=
8060
5010 RETURN
7000 REM ____________________
7005 OVER 0
7006 LET r=s↑2
7008 LET s=s-1
7009 IF s<0 THEN RETURN
7010 LET xl=EXP (.5*LN (r-s↑2))-
1
7011 LET ln=xl*2
7012 LET sl=x-xl
7013 IF sl<0 THEN LET ln=ln+sl:
LET sl=0
7015 PLOT sl,y+s: DRAW ln,0
7017 IF y-s<0 THEN GO TO 7008
7018 PLOT sl,y-s: DRAW ln,0
7019 GO TO 7008
7020 RETURN
7100 SAVE "Ariadna" LINE 1
7200 STOP
8000 REM ____________________
8010 DATA "[graphics]"
8011 DATA "[graphics]"
8012 DATA "[graphics]"
8013 DATA "[graphics]"
8014 DATA "[graphics]"
8015 DATA "[graphics]"
8016 DATA "[graphics]"
8017 DATA "[graphics]"
8018 DATA "[graphics]"
8019 DATA "[graphics]"
8020 DATA "[graphics]"
8021 DATA "[graphics]"
8022 DATA "[graphics]"
8023 DATA "[graphics]"
8024 DATA "[graphics]"
8025 DATA "[graphics]"
8026 DATA "[graphics]"
8027 DATA "[graphics]"
8028 DATA "[graphics]"
8029 DATA "[graphics]"
8030 DATA "[graphics]"
8031 DATA "[graphics]"
8050 PLOT OVER 1; INK 0;x1,y1
8051 RETURN
8060 PLOT OVER 1; INK 5;x1,y1
8061 RETURN
9000 REM ____________________
9010 FOR n=0 TO 7
9011 POKE USR "a"+n,128
9012 POKE USR "b"+n,128
9013 POKE USR "c"+n,1
9014 POKE USR "d"+n,1
9015 POKE USR "e"+n,129
9016 POKE USR "f"+n,0
9017 POKE USR "g"+n,1
9018 POKE USR "h"+n,128
9019 POKE USR "i"+n,129
9020 POKE USR "j"+n,129
9022 NEXT n
9023 POKE USR "a",255
9024 POKE USR "b"+7,255
9025 POKE USR "c",255
9026 POKE USR "d"+7,255
9027 POKE USR "f",255
9028 POKE USR "f"+7,255
9029 POKE USR "g",255
9030 POKE USR "g"+7,255
9031 POKE USR "h",255
9032 POKE USR "h"+7,255
9033 POKE USR "i"+7,255
9034 POKE USR "j",255
9035 FOR n=0 TO 7
```

Programas

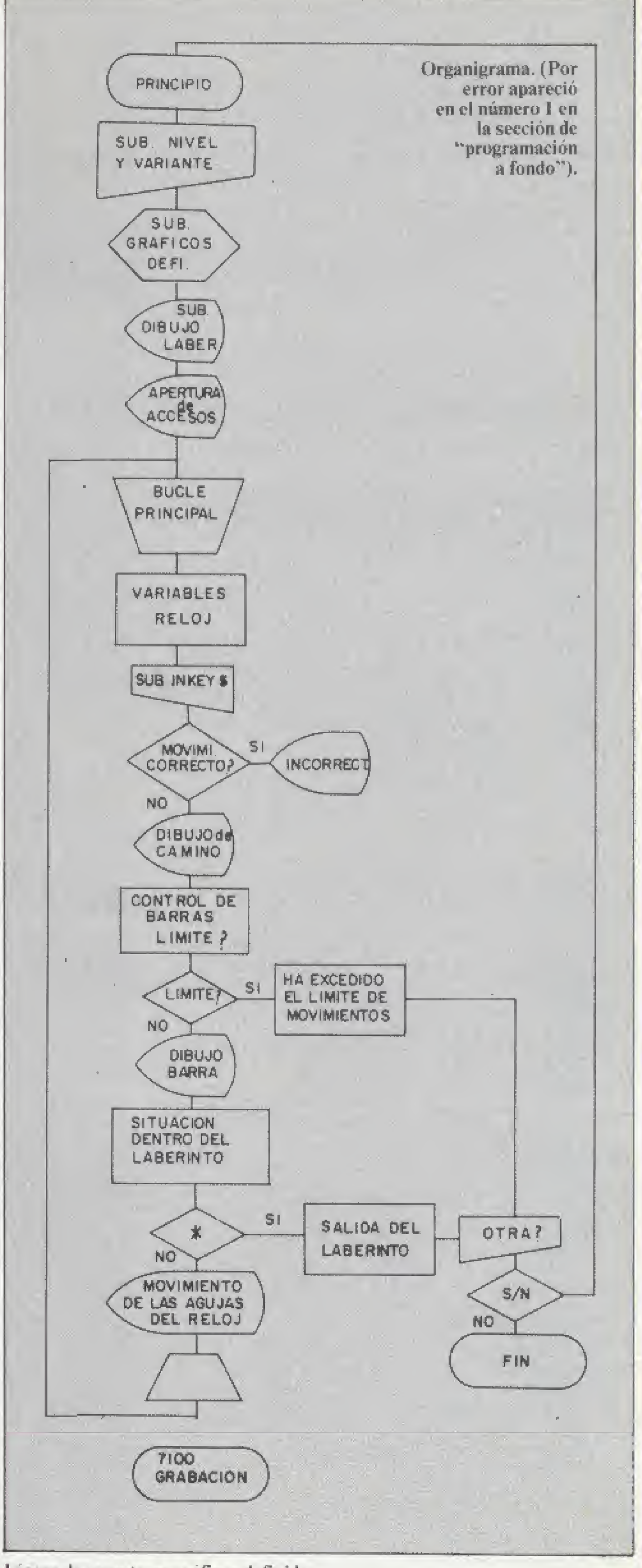

```
9037 POKE USR "l"+n,0
9038 POKE USR "m"+n,1
9039 POKE USR "n"+n,128
```



```
9040 NEXT n
9042 POKE USR "l",255
9050 RETURN
```




```
8000>REM
8010 DATA "AFFFFFFFFFFFFFLFFFFFFF
FFFFC"
8011 DATA "EAFFFFFFFFFFFCEAFFFFFF
FFFCE"
8012 DATA "EEALCHLLLFFFDEBFCACJA
CACEE"
8013 DATA "EEN  FMENGAFF FCEEBME
N_MEE"
8014 DATA "EEN_DJB_DA_FCIADEEJND
BCEEE"
8015 DATA "EEEACNFLLMHCIJBCEEB F
CBMEE"
8016 DATA "EEEEEEADENF_F FDEEJNC
IADEE"
8017 DATA "EEEEEBMA_DAFCEAFDEBDE
ADJEE"
8018 DATA "EEEEEJN LFDADEBCH_CHD
BLMEE"
8019 DATA "EEEEEBDEIAFDA CBFCEAF
CEIEE"
8020 DATA "EEEEEACNGNFF   FFMEEJ
B GEE"
8021 DATA "EEEEEN  CBFCB DAFDEB_
CIJEE"
8022 DATA "EEEEEBDIBFF_GEH_LFDAC
BFMEE"
8023 DATA "EEEEBFFFCJJACEACEALDB
FCBME"
8024 DATA "EEEEACHCEBMEB_DEBMIAF
FDJEE"
8025 DATA "EEEEEBFMBCENLFF F_FDA
FFDEE"
8026 DATA "EEEN_FCBCEEBMHCEAFFLD
AGAME"
8027 DATA "EEENCA_GENDJBFDEEAL_L
 CEEE"
8028 DATA "EEEIB_CAMNCBFLFDEEBFD
IIEEE"
8029 DATA "EEBFFFDB_DBFCIALDBFFF
FFDEE"
8030 DATA "EBFFFFFFLFFCNLMBFFFFF
FFFDE"
8031 DATA "BFFFFFFF_FGIB_DHFFFFF
FFFFD"
8050 PLOT OVER 1; INK 0;x1,y1
8051 RETURN
8060 PLOT OVER 1; INK 5;x1,y1
8061 RETURN
9000 REM
```



Organigrama. (Por error apareció en el número 1 en la sección de "programación a fondo").

Líneas de caracteres gráficos definidos.



¡¡Exclusivo para lectores de **Todospectrum**

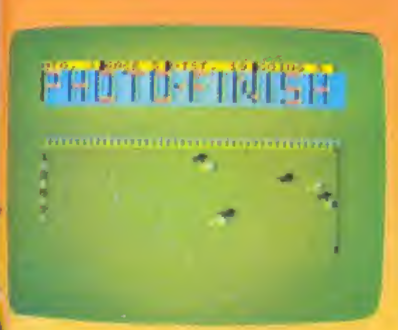


CLUB DE SOFTWARE



Ha seleccionado para usted cuatro excelentes programas de juego para su ordenador **SPECTRUM**.

Diviértase con ellos y, **además, ahorre dinero** con esta oferta única y exclusiva para los lectores de esta revista.



RACING MANAGER

Estamos en la nueva temporada de carreras. Sólo piensas en el derby. Diriges una cuadra de caballos, debes entrenarlos para el gran día. Compra y vende caballos, si es necesario, pero cuando llegue la carrera trata de dominar tu tensión. En los últimos 200 metros descubrirás si has hecho bien tu trabajo. El ordenador te ayudará, pero todo dependerá de tu buen juicio.
Precio: 1.550 ptas.



QUETZ ACOATL

Tres meses buscando el Tempolo templo de Quetzacoatl para acabar atrapado en él. Tus compañeros han sido asesinados. Tu única manera de salir es atravesando la red de pasadizos secretos construídos por los aztecas. Pero ¡atención! no incurras en 1 provoques la veganza de los dioses. Puedes consultar tu mapa, pero cada vez que lo haces se desintegra y debes comenzar de nuevo tu desesperada búsqueda de salidas. Buena suerte.
Precio 1.550 ptas.

DR. FRANKY

Solitario, en las mazmorras encantadas y oscuras de su castillo, el Dr. Franky intenta desesperadamente convertir el monstruo que él ha inventado en un ser humano. Parece simple, pero no lo es. Se trata de encontrar los frascos que contienen los productos químicos que dan la vida. Y no debes olvidar la fuerza de lo sobrenatural. Las gentes que han sido transformadas en monstruo quieren vengarse.
Precio: 1.550 ptas.



SORCERY

La Tierra ha caído en el oscurantismo. Nigromante y sus demoníacos sirvientes la han conquistado, esclavizando a los humanos. Tú eres el último de los brujos de otra época y debes viajar a Stonehenge para rescatar a otros tres brujos, a quienes necesitas para vencer a Nigromante. Se agota tu tiempo. ¿Podrá salvarlos antes de que se hayan convertido en piedra? ¿Antes que la Tierra se hunda eternamente?
Precio: 1.550 ptas.



Oferta especial 4x3

Usted puede comprar cualquiera de estos cassettes a su precio individual de **1.550** Ptas. Pero también puede adquirir el lote de 4 por sólo **4.650** Ptas. y AHORRARSE **1.550** Ptas. O sea, cuatro cassettes por el precio de tres. Aproveche esta oferta limitada.

Recorte y envíe este cupón HOY MISMO a:
INFODIS, S.A.
Bravo Murillo, 377-5.º-A
28020 MADRID

CUPON DE PEDIDO

Envíenme a mi domicilio, al precio de **1.550** Ptas. cada una, más 95 Ptas. de gastos, la cassette o cassettes que le indico a continuación marcando con una (X) en el casillero correspondiente. Queda bien entendido que pidiendo las 4 cassettes obtengo un ahorro de **1.550** Ptas., lo que me supone UNA CASSETTE GRATIS (**Sólo pagaré 4.650 Ptas. más 95 Ptas. por gastos de envío**).

☐ **RACING MANAGER** ☐ **QUETZACOATL** ☐ **DR. FRANKY** ☐ **SORCERY**

☐ Las 4 cassettes, en las ventajosas condiciones de AHORRO para mí (**4.650 Ptas.**).

El importe lo abonaré:
POR CHEQUE ☐ **CONTRA REEMBOLSO** ☐ **CON TARJETA DE CREDITO** ☐
American Express ☐ **Visa** ☐ **Interbank** ☐ Fecha de caducidad ______

Número de mi Tarjeta ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Nombre ______

Dirección ______

Ciudad ______ D.P. ______

Provincia ______

Cada envío llevará 95 Ptas. en concepto de gastos.

El misterio del boxeador que no quería pelear

En el número 2 de la revista, dos lectores escriben con la misma pega: no pueden controlar al boxeador del programa Boxeo. Una verdadera lástima, porque el juego es bastante bueno. Sin embargo el listado del programa es correcto, ¿qué ocurre entonces? Pues que, como queda dicho en el artículo "Sintonice bien su Spectrum", publicado en el número 1 de TODOSPECTRUM, existen tres versiones del ordenador. Esto es verdad....., a medias. En realidad son cuatro: la versión 1, no comercializada en España, y las versiones 2, 3 y 3B.

El *software* comercial trabaja igual en todas ellas, pero hay una curiosa diferencia. La función IN es distinta en la versión 3. Para comprobarlo no hay más que hacer PRINT IN 63486, si nuestro Spectrum es versión 2 ó 3B obtendremos el valor 255, mientras que si fuera de la versión 3 obtendríamos el valor 191.

¿Cuál es la explicación? Para entenderla abandonemos por un momento el sistema de numeración decimal y pasemos al binario. Al leer una semifila del teclado con la función IN (ver el capítulo 23 del manual), obtendremos un *byte*, formado como sabéis por 8 *bits*. Los 5 *bits* primeros de la derecha, es decir, del D0 al D4, corresponden a las 5 teclas de esa semifila: D4 es el correspondiente a la tecla más próxima a la parte central del teclado, y D0 el correspondiente a la tecla más externa. Estos *bits* toman el valor 0 si la tecla está pulsada y el 1 si no lo está.

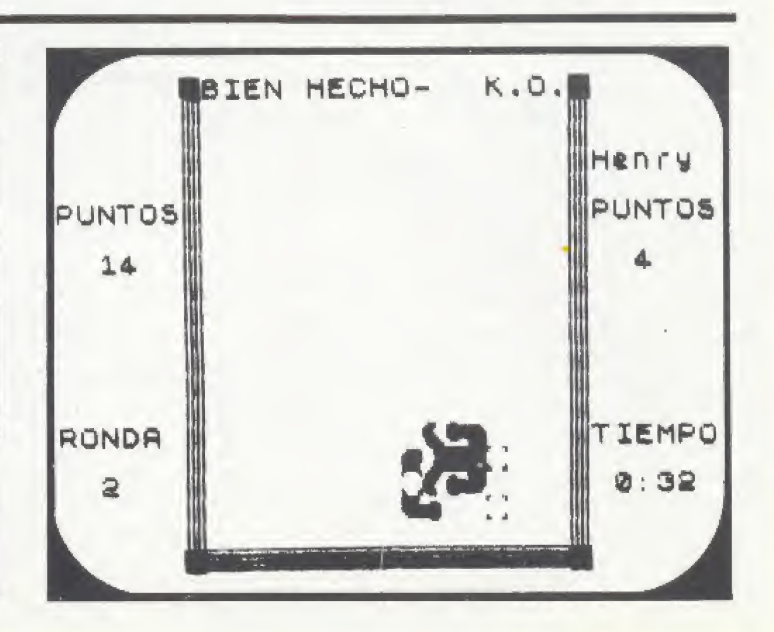
Hasta aquí todo es igual para todas las versiones. Pero ahora viene la diferencia:

Los tres *bits* restantes, esto es D5, D6 y D7, no corresponden a ninguna tecla. Pero mientras que en las versiones 2 y 3B siempre valen 1, en la versión 3 el *bit* D6 vale 0.

Resumiendo, para el que no sea amigo de complicarse la vida: comprobad si vuestro Spectrum es de la versión 3 mediante PRINT IN 63486. Si el número obtenido es el 255, es de cualquiera de las otras versiones, mientras que si es el 191 es de la versión 3. Para que el programa de Boxeo funcione en esta versión, habrá que cambiar las líneas 400, 410, 500 y 510 de la siguiente forma:

```
400 IF IN 64510 = 190
      THEN GO SUB 1000
410 IF IN 57342 = 190
      THEN GO SUB 1100
500 LET y = y +(IN 61438 = 175
      AND y < 17) - (IN 61438
      = 183 AND y > 2)
510 LET x = x +(IN 61438 = 187
      AND x < 22) - (IN 63486
      = 175 AND x > 7)
```

```
 400>IF IN 64510=190 THEN GO SUB
1000
 410 IF IN 57342=190 THEN GO SUB
1100
 450 IF g=x+1 THEN IF RND>sk THE
N GO SUB 2000
 500 LET y=y+(IN 61438=175 AND y
<17)-(IN 61438=183 AND y>2)
 510 LET x=x+(IN 61438=187 AND x
<22)-(IN 63486=175 AND x>7)
```



Como veis, lo único que hemos hecho es restar 64 en todos los valores de la función IN, que en el listado anterior eran respectivamente 254, 254, 239, 247, 251 y 239.

Con este simple cambio el poseedor de un Spectrum versión 3 verá a su boxeador perder la timidez, moviéndose por todo el ring y sacudiendo a puñetazos a su rival.

Finalmente, dar las gracias a José Ibáñez, José Salvia, Josep Marcos y Agustín Froufe, quiénes astutamente se percataron del problema y nos escribieron simpáticas cartas.

Enrique Fernández



Programas

Simulador Lógico

19

Llegó la hora de probar sus conocimientos sobre electrónica, en aquello que los "entendidos" llaman las "puertas lógicas". El tema es bastante didáctico pero sin dejar de ser divertido... ¡que de eso se trata!

El programa ilustra el funcionamiento de las puertas lógicas básicas, en las que se inspira toda la tecnología actual, incluso la de nuestro Spectrum. Y es precisamente él, quien nos explica cómo se comportan estas puertas ante la información de "ceros y unos" (tensión o no tensión) que aplicamos a las entradas, dependiendo de los valores de éstas y del tipo de puerta obtendremos un determinado valor a la salida. Seguidamente aparecerá el esquema eléctrico equivalente a la función lógica en forma de interruptores, resistencias y lámparas, incluyéndose el esquema del símbolo electrónico que representa a la puerta con los valores en sus entradas y en la salida. En el esquema se representan las entradas como interruptores, siendo "Ø" cuando están abiertos y "1" cuando están cerrados. Los signos + y – representan a los polos de la pila, las resistencias son representadas por unos rectángulos. Seguidamente el ordenador pedirá que introduzcamos los valores que queramos tener en las entradas dándonos el valor de la salida.

Autor: Juan Bautista Pérez 48K

```
  10 REM  © JUAN B.PEREZ LOPEZ
  20 REM
  30 INK 0: PAPER 7: BORDER 7: O
VER 0: BRIGHT 0: FLASH 0: CLS
  40 REM  GRAFICOS DEFINIDOS
          ------------------

  50 LET z=99: LET x=192: LET y=
255
  60 LET q=0: LET v=28: LET w=8
  70 DATA y,y,x,x,x,y,y,q,y,y,q,
q,q,y,y,q
  80 DATA y,y,3,3,3,y,y,q,127,12
7,z,z,z,z,z,z
  90 DATA z,z,z,z,z,z,z,z,z,z,z,
z,z,z,127,127
 100 DATA 24,126,126,y,y,126,126
,24,v,v,1,2,4,w,v,v
 110 DATA 4,w,211,227,195,q,q,q,
v,v,w,w,w,w,v,v
 120 DATA q,q,195,y,195,q,q,q
 130 FOR n=USR "a" TO USR "a"+87
: READ a: POKE n,a: NEXT n
 140 REM menu principal
          ---------------

 150 PRINT AT 1,8;"SIMULADOR LOG
ICO"
 160 PLOT 63,158: DRAW 129,0
 170 PLOT 63,156: DRAW 129,0
 180 PLOT 15,144: DRAW 225,0
 190 DRAW 0,-128: DRAW -225,0: D
RAW 0,128
 200 PLOT 13,146: DRAW 229,0
 210 DRAW 0,-132: DRAW -229,0: D
RAW 0,132
 220 PRINT ;AT 5,10;"elige opcio
n"
 230 PLOT 79,126: DRAW 96,0
 240 LET a$=" = puerta tipo "
 250 PRINT AT 7,4;"A"+a$+"INVERS
OR"
 260 PRINT AT 9,4;"B"+a$+"AND"
 270 PRINT AT 11,4;"C"+a$+"OR"
 280 PRINT AT 13,4;"D"+a$+"NAND"
 290 PRINT AT 15,4;"E"+a$+"NOR"
 300 PRINT AT 17,4;"F"+a$( TO 10
)+"OR EXCLUSIVA"
 310 PAUSE 0
 320 GO SUB 2210
 330 REM eleccion puertas
          -----------------

 340 LET a=PEEK 23560
 350 IF a=65 OR a=97 THEN GO TO
0440
 360 IF a=66 OR a=98 THEN GO TO
0660
 370 IF a=67 OR a=99 THEN GO TO
0950
 380 IF a=68 OR a=100 THEN GO TO
 1260
 390 IF a=69 OR a=101 THEN GO TO
 1560
 400 IF a=70 OR a=102 THEN GO TO
 1870
 410 PRINT AT 11,4;"Pulsa solo A
,B,C,D,E,F": BEEP 2,0
 420 PRINT AT 11,4;a$( TO 23)
 430 GO TO 220
 440 REM       INVERSOR
 450 REM
 460 REM dibujar esquema
 470 REM
 480 PRINT AT 18,6;"PUERTA TIPO
INVERSOR": GO SUB 2180
 490 PRINT AT 10,19;"___"
 500 CIRCLE 194,88,2: OVER 1: PL
OT 193,88: PLOT 194,88: PLOT 195
,88: OVER 0
 510 PLOT 192,88: DRAW -24,-16:
DRAW 0,32: DRAW 24,-16
 520 PRINT AT 8,19;"A     S"
 530 PRINT AT 8,3;"+  ▭": PLOT
 39,108: DRAW 8,0
 540 PLOT 71,108: DRAW 37,0: DRA
W 0,-12
 550 PLOT 108,87: DRAW 0,-12: DR
AW -67,0
 560 PLOT 91,108: DRAW 0,-12:
 570 PLOT 91,87: DRAW 0,-12
 580 PRINT AT 10,11;"█ █";AT 12,
3;"-"
 590 PRINT AT 11,10;"s";AT 11,14
;"a"
 600 REM control inversor
          -----------------
```

No se caliente la "CABEZA"

SEIKOSHA

IMPRESORAS

Nuestra calidad es "SEIKO"; nuestros precios, únicos. Si desea más información, consulte con nuestro distribuidor más cercano, o llame o escriba a:



Dirección comercial:
Av. Blasco Ibáñez, 114-116.
46022-Valencia.
Tel. (96) 372 88 89.
Télex 62220

Delegación en Cataluña:
C/ Muntaner, 60, 4, 1.
08011-Barcelona.
Tel. (93) 323 32 19.

ESTOS SON NUESTROS MODELOS:

Modelo	Velocidad	Columnas	Tipos de letra	Interface	P.V.P.
GP-50	40 cps	46	2	A-Paralelo AS-Serial S-Spectrum	A-25.900 AS-29.900 S-28.900
GP-500	50 cps	80	2	A-Paralelo AS-Serial	A-47.900 AS-49.900
GP-550	86 cps	80-136	18	A-Paralelo	A-59.900
GP-700	50 cps	80-106	3	A-Paralelo	A-89.900
BP-5200	200 cps	136-272	18	Paralelo y serial	199.000
BP-5420	420 cps	136-272	18	Paraleo y serial I-IBM PC	299.000 I-299.000

Disponemos de interfaces opcionales para todos los modelos: IBM PC, COMMODORE 64, ZX SPECTRUM, ATARI, DRAGON 64, SHRAP MZ 700, SPECTRAVIDEO, NEW BRAIN, APPLE, ETC...

Programas

```
 610 PRINT AT 21,1;" pulsa valor
 de A 1/0 (M=menu)": PAUSE 0
 620 IF INKEY$="0" THEN PRINT AT
 10,13;"/";AT 13,19;"0      1"; I
NK 5;AT 10,11;"●": BEEP .2,10
 630 IF INKEY$="1" THEN PRINT AT
 10,13;"I";AT 13,19;"1      0"; I
NK 1;AT 10,11;"●": BEEP .2,25
 640 IF INKEY$="m" OR INKEY$="M"
 THEN GO SUB 2210: GO TO 150
 650 GO TO 0590
 660 REM Puerta AND
         ----------

 670 REM
 680 REM dibujar esquema
         ---------------

 690 GO SUB 2170
 700 PRINT AT 9,19;"__";AT 11,19
;"‾‾";AT 10,24;"‾‾"
 710 PLOT 167,103: DPAW 0,-30: D
RAW 10,0: DRAW 0,30,3: DRAW -10,
0
 720 PRINT AT 8,3;"+    ■  ■";AT
12,3;"-";AT 10,13;"■";AT 18,8;"P
UERTA TIPO AND"
 730 PLOT 39,108: DRAW 68,0: DRA
W 0,-35: DRAW -68,0
 740 PRINT AT 9,6;"a  b";AT 10,1
2;"s"
 750 PRINT AT 8,19;"A";AT 13,19;
"B";AT 9,25;"S"
 760 REM Conrol AND
 770 PRINT AT 21,1;" pulsa valor
 de A 1/0 (M=menu)": PAUSE 0
 780 PRINT AT 10,27;" ";AT 11,17
;" "
 790 IF INKEY$="m" OR INKEY$="M"
 THEN GO SUB 2210: GO TO 150
 800 LET a$=INKEY$
 810 IF a$<>"1" AND a$<>"0" THEN
 GO TO 0790
 820 LET a=VAL a$: PRINT AT 9,17
;a: IF a=0 THEN PRINT AT 8,7;"↦"
: GO TO 0840
 830 PRINT AT 8,7;"↦"
 840 BEEP .2,20
 850 PRINT AT 21,1;" pulsa valor
 de B 1/0          ": PAUSE 0
 860 LET b$=INKEY$
 870 IF b$<>"1" AND b$<>"0" THEN
 GO TO 0860
 880 LET b=VAL b$: PRINT AT 11,1
7;b: IF b=0 THEN PRINT AT 8,10;"
↦": GO TO 0900
 890 PRINT AT 8,10;"↦"
 900 LET s=a AND b: PRINT AT 10,
27;s
 910 BEEP .2,15
 920 IF s=0 THEN PRINT INK 1;AT
10,13;"●": GO TO 0940
 930 PRINT INK 5;AT 10,13;"●"
 940 PAUSE 3: GO TO 0760
 950 REM Puerta OR
         ---------

 960 REM
 970 REM dibujar esquema
         ---------------

 980 GO SUB 2170
 990 PRINT AT 9,19;"__";AT 11,19
;"‾‾";AT 10,24;"‾‾"
1000 PLOT 167,103: DRAW 0,-30,-1
.5: DRAW 10,0: DRAW 0,30,3: DRAW
 -10,0
1010 PRINT AT 8,19;"A";AT 13,19;
"B";AT 9,25;"S"
1020 PRINT AT 7,9;"a";AT 11,9;"b
";AT 11,12;"s"
1030 PRINT AT 8,3;"+        ■";AT
10,10;"■";AT 10,13;"■";AT 12,3;"
-"
1040 PLOT 39,108: DRAW 68,0: DRA
W 0,-35: DRAW -68,0
1050 PLOT 75,108: DRAW 0,-16: DR
AW 16,0: DRAW 0,16
1060 PRINT AT 18,8;"PUERTA TIPO
OR"
1070 REM control OR
         ----------

1080 PRINT AT 21,1;" pulsa valor
 de A 1/0 (M=menu)": PAUSE 0
1090 PRINT AT 10,27;" ";AT 11,17
;" "
1100 IF INKEY$="m" OR INKEY$="M"
 THEN GO SUB 2210: GO TO 150
1110 LET a$=INKEY$
1120 IF a$<>"1" AND a$<>"0" THEN
 GO TO 1110
1130 LET a=VAL a$: PRINT AT 9,17
;a: IF a=0 THEN PRINT AT 8,10;"↦
```



19

```
": GO TO 1150
1140 PRINT AT 8,10;"↦"
1150 BEEP .2,25
1160 PRINT AT 21,1;" pulsa valor
 de B 1/0         ": PAUSE 0
1170 LET b$=INKEY$
1180 IF b$<>"1" AND b$<>"0" THEN
 GO TO 1160
1190 LET b=VAL b$: PRINT AT 11,1
7;b: IF b=0 THEN PRINT AT 10,10;
"↤": GO TO 1210
1200 PRINT AT 10,10;"↦"
1210 BEEP .2,15
1220 LET s=a OR b: PRINT AT 10,2
7;s
1230 IF s=0 THEN PRINT INK 1;AT
10,13;"●": GO TO 1250
1240 PRINT INK 5;AT 10,13;"●"
1250 PAUSE 3: GO TO 1080
1260 REM puerta NAND
         -----------

1270 REM
1280 REM dibujar esquema
         ---------------

1290 GO SUB 2170
1300 PRINT AT 9,19;"__";AT 11,19
;"__";AT 10,24;"__"
1310 CIRCLE 194,88,2: OVER 1: PL
OT 193,88: PLOT 194,88: PLOT 195
,88: OVER 0
1320 PLOT 167,103: DRAW 0,-30: D
RAW 10,0: DRAW 0,30,3: DRAW -10,
0
1330 PRINT AT 8,19;"A";AT 13,19;
"B";AT 9,25;"S"
1340 PLOT 39,108: DRAW 68,0: DRA
W 0,-32: DRAW -68,0: PLOT 84,108
: DRAW 0,-32
1350 PRINT AT 8,3;"+";AT 8,6;"▀▀
▀";AT 8,12;"■";AT 12,12;"■";AT 1
2,3;"-";AT 10,10;"■"
1360 PRINT AT 18,8;"PUERTA TIPO
NAND";AT 7,13;"a";AT 13,13;"b";A
T 10,9;"s"
1370 REM control NAND
         ------------

1380 PRINT AT 21,1;" pulsa valor
 de A 1/0 (M=menu)": PAUSE 0
1390 PRINT AT 10,27;" ";AT 11,17
;" "
1400 IF INKEY$="m" OR INKEY$="M"
 THEN GO SUB 2210: GO TO 150
1410 LET a$=INKEY$
1420 IF a$<>"1" AND a$<>"0" THEN
 GO TO 1410
1430 LET a=VAL a$: PRINT AT 9,17
;a: IF a=0 THEN PRINT AT 8,12;"↤
": GO TO 1450
1440 PRINT AT 8,12;"↦"
1450 BEEP .2,20
1460 PRINT AT 21,1;" pulsa valor
 de B 1/0         ": PAUSE 0
1470 LET b$=INKEY$
1480 IF b$<>"1" AND b$<>"0" THEN
 GO TO 1460
1490 LET b=VAL b$: PRINT AT 11,1
7;b: IF b=0 THEN PRINT AT 12,12;
"↤": GO TO 1510
1500 PRINT AT 12,12;"↦"
1510 BEEP .2,10
1520 LET s=NOT (a AND b): PRINT
AT 10,27;s
1530 IF s=0 THEN PRINT INK 1;AT
10,10;"●": GO TO 1550
1540 PRINT INK 5;AT 10,10;"●"
1550 PAUSE 3: GO TO 1380
1560 REM Puerta NOR
         ----------

1570 REM
1580 REM dibujar esquema
         ---------------

1590 GO SUB 2170
1600 PRINT AT 9,19;"__";AT 11,19
;"__";AT 10,24;"__"
1610 PLOT 167,103: DRAW 0,-30,-1
.5: DRAW 10,0: DRAW 0,30,3: DRAW
 -10,0
1620 PRINT AT 8,19;"A";AT 13,19;
"B";AT 9,25;"S"
1630 CIRCLE 194,88,2: OVER 1: PL
OT 193,88: PLOT 194,88: PLOT 195
,88: OVER 0
1640 PRINT AT 11,8;"a";AT 11,10;
"b";AT 11,12;"s"
1650 PLOT 39,108: DRAW 68,0: DRA
W 0,-32: DRAW -68,0: PLOT 76,108
: DRAW 0,-32: PLOT 92,108: DRAW
0,-32
1660 PRINT AT 8,3;"+";AT 8,6;"▀▀
▀";AT 10,11;"■";AT 10,13;"■";AT
12,3;"-";AT 10,9;"■"
1670 PRINT AT 18,8;"PUERTA TIPO
NOR"
1680 REM Control NOR
         -----------

1690 PRINT AT 21,1;" pulsa valor
 de A 1/0 (M=menu)": PAUSE 0
1700 PRINT AT 10,27;" ";AT 11,17
;" "
1710 IF INKEY$="m" OR INKEY$="M"
 THEN GO SUB 2210: GO TO 150
1720 LET a$=INKEY$
1730 IF a$<>"1" AND a$<>"0" THEN
 GO TO 1720
1740 LET a=VAL a$: PRINT AT 9,17
;a: IF a=0 THEN PRINT AT 10,9;"/
": GO TO 1760
1750 PRINT AT 10,9;"I"
1760 BEEP .2,25
1770 PRINT AT 21,1;" pulsa valor
 de B 1/0         ": PAUSE 0
1780 LET b$=INKEY$
1790 IF b$<>"1" AND b$<>"0" THEN
 GO TO 1780
1800 LET b=VAL b$: PRINT AT 11,1
7;b: IF b=0 THEN PRINT AT 10,11;
"/": GO TO 1820
1810 PRINT AT 10,11;"I"
1820 BEEP .2,15
1830 LET s=NOT (a OR b): PRINT A
T 10,27;s
1840 IF s=0 THEN PRINT INK 1;AT
10,13;"●": GO TO 1680
1850 PRINT INK 5;AT 10,13;"●"
1860 PAUSE 3: GO TO 1680
1870 REM Puerta OR exclusiva
         ------------------

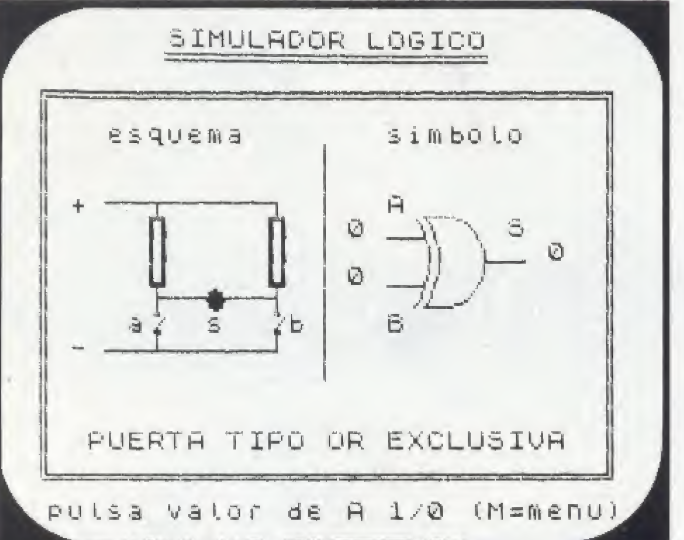
1880 REM
1890 REM dibujar esquema
         ---------------

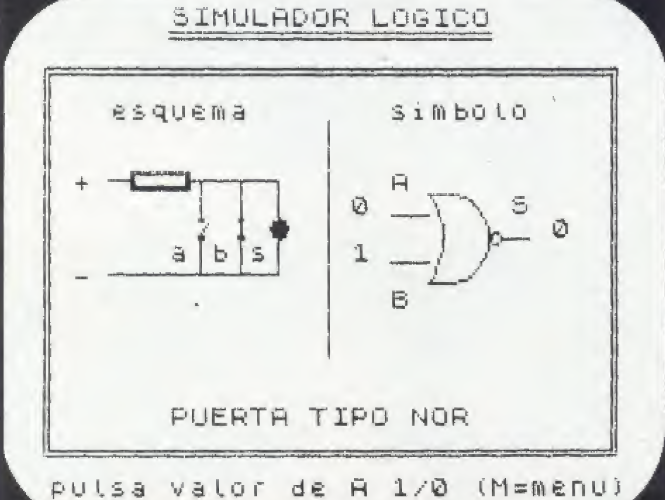
1900 GO SUB 2170
1910 PRINT AT 9,19;"__";AT 11,19
```

Programas

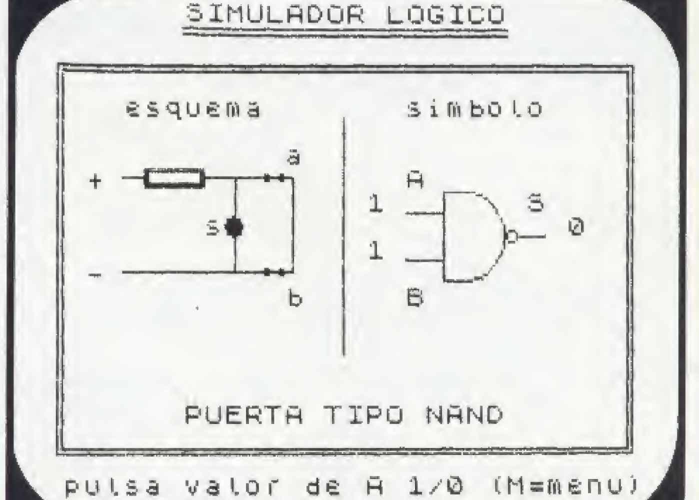

```
;"__";AT 10,24;"__"
1920 PLOT 167,103: DRAW 0,-30,-1
.5: DRAW 10,0: DRAW 0,30,3: DRAW
 -10,0: PLOT 163,103: DRAW 0,-30
,-1.5
1930 PRINT AT 8,19;"A";AT 13,19;
"B";AT 9,25;"S"
1940 PRINT AT 8,3;"+";AT 12,10;"
■";AT 13,6;"a■  s  ■b";AT 14,3;"
-"
1950 PLOT 39,108: DRAW 69,0: DRA
W 0,-50: DRAW -69,0: PLOT 60,108
: DRAW 0,-50: PLOT 60,76: DRAW 4
5,0
1960 PRINT AT 18,4;"PUERTA TIPO
OR EXCLUSIVA"
1970 PRINT AT 9,7;"∩      ∩";AT 1
0,7;"‖      ‖";AT 11,7;"∪      ∪"
1980 REM control or ex.
         --------------

1990 PRINT AT 21,1;" pulsa valor
 de A 1/0 (M=menu)": PAUSE 0
2000 PRINT AT 10,27;" ";AT 11,17
;" "
2010 IF INKEY$="m" OR INKEY$="M"
 THEN GO SUB 2210: GO TO 150
```






```
2020 LET a$=INKEY$
2030 IF a$<>"1" AND a$<>"0" THEN
 GO TO 2020
2040 LET a=VAL a$: PRINT AT 9,17
;a: IF a=0 THEN PRINT AT 13,7;"/
": GO TO 2060
2050 PRINT AT 13,7;"I"
2060 BEEP .2,25
2070 PRINT AT 21,1;" pulsa valor
 de B 1/0           ": PAUSE 0
2080 LET b$=INKEY$
2090 IF b$<>"1" AND b$<>"0" THEN
 GO TO 2080
2100 LET b=VAL b$: PRINT AT 11,1
7;b: IF b=0 THEN PRINT AT 13,13;
"/": GO TO 2120
2110 PRINT AT 13,13;"I"
2120 BEEP .2,15
2130 LET s=(a OR b) AND (NOT (a
AND b)): PRINT AT 10,27;s
2140 IF s=0 THEN PRINT INK 1;AT
12,10;"●": GO TO 2160
2150 PRINT INK 5;AT 12,10;"●"
2160 PAUSE 3: GO TO 1990
2170 REM PRINT PRESENTACION
2180 PLOT 127,128: DRAW 0,-80
2190 PRINT AT 5,5;"esquema";AT 5
,19;"simbolo"
2200 RETURN
2210 REM borrado ventana
2220 LET a$="
"
2230 FOR n=4 TO 18: PRINT AT n,2
;a$: BEEP .006,n*2: NEXT n
2240 PRINT AT 21,2;a$+" ": RETUR
N
```



GRAFICOS

A B C D E F G H I J K

⊏ = ⊐ ∩ ‖ ∪ ● / ⌿ I ⊢

SOFTWARE PROFESIONAL PARA ZX-SPECTRUM

64 COLUMNAS

CONTEXT V-6 VENTAMATIC SOFT
Copyright O.Domingo 1984

NUEVO

MENU

-1- IMPRESION N.
-2- COD.Impresion
-3- INTERFACE
-4- CHR$ Graficos
-5- MEMORIA EXTERNA
-6- TEXTOS Pers.
-7- CATALOGO
-8- EDITOR
-9- Camb.en INFO
-0- IMPR.con FICHA

- DEFINIR FORMATO FICHAS
- ORDENAR ARCHIVO
- ENTRAR FICHAS
- SELECCIONAR INFORMACION
- VER FICHAS SELECCIONADAS
- ANULAR FICHAS SELECCIONADAS
- LISTAR FICHAS SELECCIONADAS IMPRESORA
- LISTAR INFORMACION PAGINADA IMPRESORA
- SAVE PROGR.+ ARCHIVO
- SAVE SOLO ARCHIVO
- DESCARGAR FICHAS SELECCIONADAS
- LOAD ARCHIVO
- MERGE ARCHIVO
- CAMBIAR COLORES

ARCHIVO:Prueba Operacion:MENU Fin:EDIT Caps:
Hay 0 fichas Seleccionadas: 0 Mem.Libre:30000

NUEVO

CONTEXT V.6 (*) — Procesador de textos 64 columnas. Caracteres españoles. Cartas personalizadas. ¡Sensacional!

P.V.P. 4.000,— Ptas.

CONTABILIDAD PERSONAL (*) — Contabilidad doméstica o de pequeño negocio. 64 columnas. **P.V.P. 2.500,— Ptas.**

HISOF PASCAL.— El único compilador de Pascal creado para el Spectrum.

P.V.P. 3.500,— Ptas.

SITI (*) — Sistema integrado de tratamiento de información. Base de datos con posibilidad de cálculos complejos y ficheros auxiliares. **P.V.P. 4.000.— PTAS.**

BETA BASIC.— Más de 50 nuevos comandos de Basic para tu Spectrum.

P V.P. 3.000,— PTAS.

HISOF DEVPAC (*).— Conjunto de dos programas, ensamblador y desensamblador de código máquina. Impresionantes. (GENS 3M2 y MONS 3M2). **P.V.P. 3.500,— PTAS.**

CYRUS IS CHESS— El mejor ajedrez.
Campeón de Europa de 1981.
P.V.P. 3.000,— PTAS.

* Compatible con Microdive y/o cassette y cualquier impresora e interface de impresora

---

GRAFPAD

TABLETA DIGITALIZADORA PROFESIONAL
PARA ZX-SPECTRUM 48 K.

Los mejores dibujos realizados cómodamente. Trabaja hasta 3 pantallas a la vez. Incluye programa de funcionamiento y manual completo en castellano.

P.V.P. 39.000 PTAS.

AHORA EN ESPAÑA

TIENDA:
C/. Córcega, 89 - Entlo.
08029-BARCELONA
Tel. (91) 230 97 90

VENTAMATIC 

PEDIDOS POR CORREO:
Avda. de Rhode, 253
ROSAS (GERONA)
Tel. (972) 25 56 16



(20)

La Bolsa

Con este programa podrá jugar tranquilamente a "La Bolsa" y decimos tranquilamente porque no perderá grandes sumas de dinero y, por supuesto, no será necesario que se lance por la ventana si no tuvo suerte.

El juego tiene cuatro niveles de dificultad, lo que determina cuántas acciones cambiarán de valor independientemente de la información que se da a los jugadores. Dependiendo del número de jugadores, deberá elegir su nivel cuidadosamente. Así, por ejemplo, si juega solo le aconsejamos que elija el nivel 4; para dos es conveniente el nivel 3; con tres es mejor el nivel 2; y para cuatro o más jugadores es aconsejable el nivel 1.

En el mercado existen siete compañías en las que podrá invertir o especular, especialmente con los rumores de que le informa el ordenador. Las acciones de Akran, Belso, Cobra y Dobro no fluctúan demasiado, mientras que las de Elsa y Finet presentan oscilaciones mayores. Con las acciones de la Compañía Bchip será difícil que pierda dinero, pero más difícil aún será que se convierta en un Rockefeller.

Anímese y ponga en juego algo más que su imaginación.

Autor: P. Y. 16K

```
   1 REM La Bolsa
  10 INK 0: PAPER 7: BORDER 7: C
LS : POKE 23609,50: POKE 23658,8
  15 PRINT AT 10,10; PAPER 1; IN
K 6;"LA BOLSA"
  20 LET BRIBEATTEMPT=0: LET GAM
ETURN=1
  21 PRINT #1;"Nivel (1-4) ?";:
GO SUB 9100: LET LEVEL=VAL K$: P
RINT #1;CHR$ 8;K$
  22 PRINT #1;"Numero de Jugador
es ?";: GO SUB 9100: LET NUMOFPL
AYERS=VAL K$: PRINT #1;CHR$ 8;K$
  23 PRINT #1;"Numero de Jugadas
 ?";: GO SUB 9100: LET GAMELENGT
H=VAL K$: PRINT #1;CHR$ 8;K$
  25 RANDOMIZE : GO SUB 8000
  30 GO TO 100
  42 PRINT AT 0,0; INVERSE 1;"AC
C.   VALOE DISPO  MANT  S TEND"
  43 LET L=1
  45 FOR R=3 TO 15 STEP 2: PRINT
 AT R,0;N$(L);AT R,7;"'";AT R,9;
S(1,L);AT R,29;S(3,L): IF S(4,L)
<>1 THEN PRINT AT R,27;S(5,L): P
RINT AT R,26;"S": GO TO 48
  46 PRINT AT R,26;"N"
  48 LET L=L+1: NEXT R
  50 RETURN
 100 REM Bucle principal.
```

```
 105 GO SUB 40
 110 FOR N=1 TO NUMOFPLAYERS: GO
 SUB 120: GO SUB 150: GO TO 210
 120 PRINT AT 1,10; PAPER 5;"Jug
ador  ";N
 125 LET HOLD=1
 130 FOR R=3 TO 15 STEP 2: PRINT
 AT R,14;S(2,HOLD);AT R,21;P(HOL
D,N);"  ": LET HOLD=HOLD+1: NEXT
 R
 140 PRINT AT 16,26;"      ": PRI
NT AT 16,2; PAPER 5;"Disponible
en Jugada  ";GAMETURN;"     ";P(8,
N): RETURN
 150 PRINT AT 18,0; PAPER 6;"Rum
ores": FOR R=18 TO 20: GO SUB 60
00: NEXT R
 152 PRINT AT 21,10;"
 "
 155 LET BRIBE=10+5*INT (RND*5)
 160 PRINT #1;AT 1,0;"Soborno ";
BRIBE;"?     ": GO SUB 9100: LET
Y$=K$
 165 IF Y$<>"S" AND Y$<>"SI" AND
 Y$<>"N" AND Y$<>"NO" THEN GO TO
 160
 170 IF Y$="S" OR Y$="SI" THEN L
ET BRIBEATTEMPT=1: LET R=21: GO
SUB 6000
 180 LET BRIBEATTEMPT=0
 190 RETURN
 210 GO SUB 5000
 220 NEXT N
 225 GO SUB 4000: FOR N=1 TO 6:
IF S(5,N)<>0 THEN LET S(4,N)=2
 227 NEXT N
 230 LET S(3,7)=INT (RND*3): REM
 BCHIP INCREASE (IF ANY)
 240 FOR N=1 TO 7: LET S(1,N)=S(
1,N)+S(3,N): IF S(1,N)<=0 THEN G
O SUB 7000
 245 IF S(1,N)>40 THEN LET S(1,N
)=40
 250 NEXT N
 260 FOR N=1 TO 6: IF S(4,N)=2 T
HEN LET S(5,N)=S(5,N)-1: IF S(5,
N)=0 THEN LET S(4,N)=1
 270 NEXT N
 280 GO SUB 40: PRINT AT 1,10; P
APER 5;"SITUACION": PRINT AT 17,
3; PAPER 6;"Pulse cualquier tecl
a": PAUSE 0
 300 LET GAMETURN=GAMETURN+1: IF
 GAMETURN=GAMELENGTH THEN PRINT
AT 19,10; INVERSE 1;"Ultima juga
da"
 310 IF GAMETURN>GAMELENGTH THEN
 GO TO 9000
 320 FOR N=1 TO 6: LET S(3,N)=IN
T (0.5+(.5*S(3,N))): IF S(1,N)=4
0 THEN LET S(3,N)=0
 330 NEXT N: LET S(3,7)=0: GO TO
 105
4000 REM Rumores
4010 CLS : PRINT AT 3,0; PAPER 5
;"Rumores desconocidos para los
   jugadores."
4020 FOR R=7 TO 6+LEVEL: LET RAN
1=1+INT (RND*47): GO SUB 6012: N
EXT R
4025 PAUSE 150
4030 RETURN
5000 REM Jugadores
5010 PRINT AT 17,0; PAPER 2; INK
 7;" Presione ""E"" para finaliz
ar."
5020 PRINT #1;AT 0,0;"
                  Vende (S)
o Compra (B)            ": GO SUB 91
00: LET Y$=K$
5025 IF Y$="E" THEN PRINT #1;AT
1,0;"No se retire!!
 ": PAUSE 150: RETURN
5027 IF Y$="S" OR Y$="B" THEN GO
 TO 5030
5029 GO TO 5020
5030 INPUT "Numero de acciones "
;NUM
5040 INPUT "Nombre empresa "; LI
NE C$: FOR L=1 TO 7: IF C$=N$(L)
 THEN GO TO 5050
5045 NEXT L: PRINT #1;AT 0,0;"No
 existe la empresa ";C$," -Prueb
e de nuevo": PAUSE 150: GO TO 50
20
5050 IF S(4,L)<>1 THEN PRINT #1;
AT 1,0;"Acc. suspendidas    ": PA
USE 150: GO TO 5020
5055 IF Y$="S" THEN GO TO 5100
5057 IF Y$<>"B" THEN GO TO 5020
5060 IF NUM>S(2,L) THEN PRINT #1
;AT 0,0;"No hay        ";NUM," a
cciones": PAUSE 150: GO TO 5020
5070 IF P(8,N)<S(1,L)*NUM THEN P
RINT #1;AT 0,0;"No tiene suficie
nte dinero": PAUSE 150: GO TO 50
20
5080 LET S(2,L)=S(2,L)-NUM: LET
P(L,N)=P(L,N)+NUM: LET P(8,N)=P(
8,N)-(S(1,L)*NUM): BEEP .25,0: P
RINT #1;AT 0,0;NUM;" Acciones ";
C$;" costaron    -  '";S(1,L)*NU
M: PAUSE 200
5090 PRINT AT 1+L*2,14;S(2,L);"
 ": PRINT AT 1+L*2,21;P(L,N);"
": GO SUB 140: GO TO 5020
5100 IF NUM>P(L,N) THEN PRINT #1
;AT 0,0;"No dispone de ";NUM;" a
cciones de ",C$;: PAUSE 150: GO
TO 5020
5110 LET P(8,N)=P(8,N)+NUM*S(1,L
): LET S(2,L)=S(2,L)+NUM: LET P(
L,N)=P(L,N)-NUM: PRINT #1;AT 0,0
;NUM;" Acciones ";C$;" vendidas
por '";S(1,L)*NUM: PAUSE 200: GO
 TO 5090
6000 REM Informacion a jugadores
.
6005 IF BRIBEATTEMPT=1 THEN LET
P(8,N)=P(8,N)-BRIBE: GO SUB 140:
 IF RND>.6 THEN GO TO 6180
6010 LET RAN1=1+INT (RND*49)
6012 IF RAN1>32 THEN GO TO 6070
6015 LET RAN1=1+INT (RND*8)
6020 IF M(RAN1)>0 THEN LET W$="A
r.": GO TO 6040
6030 LET W$="Ab."
6040 LET RAN2=1+INT (RND*4)
6050 LET S(3,RAN2)=S(3,RAN2)+M(R
AN1)
6060 PRINT AT R,10;N$(RAN2);" ";
W$;" ";ABS M(RAN1);"        ":
RETURN
6070 IF RAN1>42 THEN GO TO 6120
6080 LET RAN1=9+INT (RND*5)
6090 IF M(RAN1)>0 THEN LET W$="A
r.": GO TO 6110
6100 LET W$="Ab."
```

Programas

20

```
6110 LET RAN2=5+INT (RND*2): GO
TO 6045
6120 IF RAN1=43 THEN FOR L=1 TO
6: IF S(4,L)=1 THEN LET S(3,L)=S
(3,L)+2: NEXT L: PRINT AT R,10;"
Todas '2 Ar.              ": RETURN
6130 IF RAN1=44 THEN FOR L=1 TO
6: IF S(4,L)=1 THEN LET S(3,L)=S
(3,L)-2: NEXT L: PRINT AT R,10;"
Todas '2 Ab.              ": RETURN
6140 IF RAN1>=48 THEN GO TO 6180
6150 LET RAN2=1+INT (RND*6): LET
 S(5,RAN2)=2+INT (RND*3): PRINT
AT R,10;N$(RAN2);" Cancelado
 ": RETURN
6160 PRINT AT R,10;"Ascenso canc
elado ": INPUT "Empresa ";Y$: FO
R L=1 TO 6: IF Y$=N$(L) THEN GO
TO 6175
6170 NEXT L: PRINT AT R,27;"Ning
una": RETURN
6175 LET S(4,L)=1: LET S(5,L)=0:
 PRINT AT R,27;Y$: PRINT AT 1+L*
2,26;"N ": RETURN
6180 LET RAN1=1+INT (RND*4)
6190 LET P(8,N)=P(8,N)-50*RAN1
6200 PRINT AT R,10;"Multa '";50*
RAN1;" por soborno": PAUSE 100:
GO SUB 140
6220 RETURN
7000 REM Bancarrota
7010 FOR L=1 TO NUMOFPLAYERS: LE
T P(N,L)=0: LET P(N,L)=0: NEXT L
7020 LET S(1,N)=15: LET S(3,N)=0
: LET S(4,N)=1: LET S(2,N)=100:
LET S(5,N)=0
7030 BEEP .4,0: BEEP .3,0: BEEP
.2,0: BEEP .4,0: BEEP .4,3: BEEP
 .2,2: BEEP .3,2: BEEP .2,0: BEE
P .3,0: BEEP .2,-1: BEEP .4,0: P
RINT AT 18,0;"Perdidas las accio
nes de ";N$(N);" Bancarrota": PA
USE 130
7040 RETURN
8000 REM Inicializacion.
8010 DIM M(14): DIM N$(7,5)
8020 FOR N=1 TO 14: READ M(N): N
EXT N
8030 DATA -4,-3,-2,-1,1,2,3,4,6,
4,2,-10,-6,0
8040 FOR N=1 TO 7: READ N$(N): N
EXT N
8050 DATA "AKRAN","BELSO","COBRA
","DOBRO","ELSAT","FINET","BCHIP
"

8110 DIM S(5,7): DIM P(8,NUMOFPL
AYERS): DIM W$(4)
8120 FOR L=1 TO 7: LET N=1
8130 LET S(N,L)=15
8140 LET S(N+1,L)=100
8150 LET S(N+2,L)=0
8160 LET S(N+3,L)=1
8165 LET S(N+4,L)=0
8170 NEXT L
8175 LET S(1,7)=10
8180 FOR L=1 TO 7: FOR N=1 TO NU
MOFPLAYERS
8190 LET P(L,N)=0: NEXT N
8200 NEXT L
8210 FOR N=1 TO NUMOFPLAYERS: LE
T P(8,N)=1000: NEXT N
8220 RETURN
9000 REM Fin del juego.
9005 CLS : PRINT AT 1,10; PAPER
6;"FIN DEL JUEGO"
9010 LET R=4
9020 FOR N=1 TO NUMOFPLAYERS
9030 FOR L=1 TO 7: LET P(8,N)=P(
8,N)+S(1,L)*P(L,N): NEXT L
9040 PRINT AT R,0;"JUGADOR ";N;"
 TOTALIZO ' ";P(8,N);: LET R=R+4
9050 NEXT N
9060 STOP
9100 REM Entrada.
9110 PAUSE 1: PAUSE 0
9120 LET K$=INKEY$
9130 RETURN
```

```
          Jugador  1

AKRAN   ' 15   100    0    N  0

BELSO   ' 15   100    0    N  0

COBRA   ' 15   100    0    N  0

DOBRO   ' 15   100    0    N  0

ELSAT   ' 15   100    0    N  0

FINET   ' 15   100    0    N  0

BCHIP   ' 10   100    0    N  0
 Disponible en jugada 1     1000

Rumores    AKRAN Ar.  2
           Ascenso cancelado
```



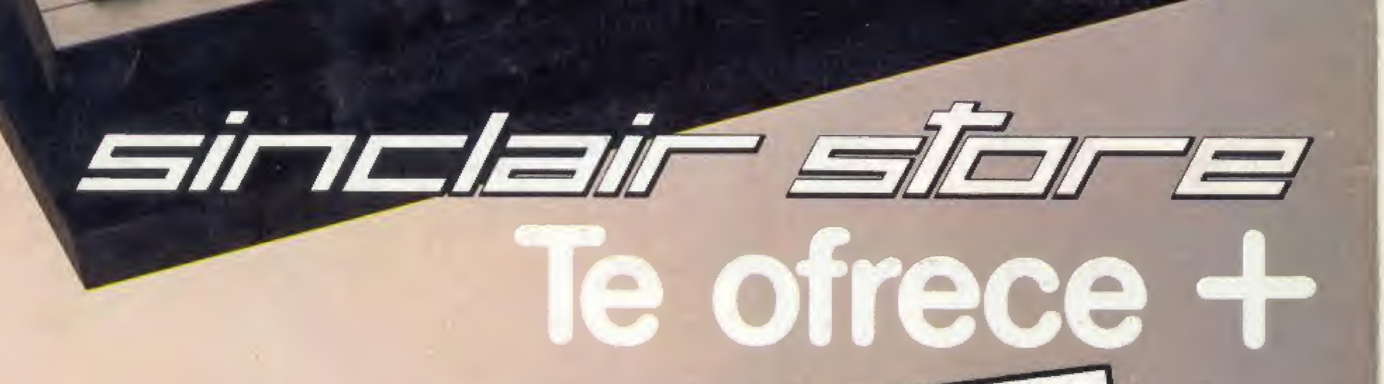
En el próximo número más...

- CONSOLA PARA EL SPECTRUM. Constrúyaselo usted mismo. Ordenador, impresora y amplificador de sonido en perfecta unión.
- PASCAL. Se inicia la serie del lenguaje más prometedor en el mundo de la informática, ya disponible para el Spectrum.
- CALENDARIO. Convierta su Spectrum en una agenda del próximo año y obtenga un bonito calendario.
- EL CUBO DE RUBIC. Tortuoso juego bidimensional.

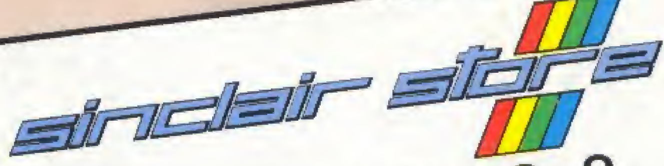


Programas/
ns/Montajes/Còdigo Màquina
YA ESTA A LA VENTA
AÑO I - No. 12 / NOVIEMBRE - 84 - 250 Ptas.
ZX
REVISTA PARA LOS USUARIOS
DE ORDENADORES SINCLAIR
Novedad
PRESENTACION DEL
SPECTRUM PLUS
CURSO DE FORTH

ZX Spectrum +



sinclair store
Te ofrece +



BRAVO MURILLO, 2
(aparc. gratuíto en c/. Magallanes, 1)
Tel. 446 62 31
DIEGO DE LEON, 25 - Tel. 261 88 01
MADRID